

LAMPIRAN

Lampiran 1. RPS Mata Kuliah Praktik Mikrokontroler

 <p>KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI UNIVERSITAS NEGERI YOGYAKARTA FAKULTAS TEKNIK</p>		 <p>RENCANA PEMBELAJARAN SEMESTER</p>			<small>NO.:RPS/MEK/62/18/2014</small>	<small>SEM: III</small>	<small>SKS: 2P</small>	<small>Revisi: 01</small>	<small>Tanggal 28 Agustus 2015</small>	<small>Certificate No: QSC 0059</small>
PROGRAM STUDI	: PENDIDIKAN TEKNIK MEKATRONIKA									
MATA KULIAH	: PRAKTIK MIKROKONTROLER									
DOSEN PENGAMPU	: TIM									
I. DESKRIPSI MATA KULIAH Perkuliahan Praktik Mikrokontroler mengembangkan pemikiran dan mempraktikan aplikasi mikrokontroler secara kontekstual. Materi kuliah terdiri perangkat keras(hardware) dan lunak(software), dan aplikasi mikrokontroler. Perangkat keras berupa materi sistem minimum mikrokontroler, sedangkan perangkat lunak berupa materi bahasa pemrograman mikrokontroler. Aplikasi yang dipraktikan terdiri dari rangkaian dan pemrograman I/O (Input-Output), timer-counter, interupsi, 7 segment, keypad, LCD(Liquid Crystal Display), Motor stepper, pengaturan PWM motor DC, komunikasi serial. Perkuliahan dilaksanakan dengan pendekatan <i>Problem Based learning</i> . Penilaian berbasis kompetensi melibatkan partisipasi aktif, dan produk hasil perangkat keras dan lunak setiap setiap kompetensi..										
II. CAPAIAN PEMBELAJARAN MATA KULIAH 1. Beritaqawa kepada Tuhan YME dan mampu menunjukkan sikap religius dan berkarakter, 2. Mahasiswa berpartisipasi aktif bertanggungjawab, dan memiliki motivasi mengembangkan diri, 3. Mahasiswa mampu memahami dan membuat rangkaian sistem minimum mikrokontroler dan aplikasinya 4. Mahasiswa mampu memahami dan membuat program mikrokontroler dengan software code-vision										
Dibuat oleh: Herlambang SP	Dilarang memperbarui yak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik, Universitas Negeri Yogyakarta								Ketua Prodi :	Dikeniksa oleh:

5. Mahasiswa mampu memahami dan membuat program berbagai aplikasi mikrokontroler

III. MATRIK RENCANA PEMBELAJARAN

Peremuan ke	Capaian Pembelajaran	Bahan Kajian	Model/Metode Pembelajaran	Pengalaman Belajar	Indikator Penilaian	Teknik Penilaian	Bobot Tuguhan	Waktu	Referensi
1	2	3	4	5	6	7	8	9	10
1	Pengenalan Program Code_Vision	Software Code-Vision	Ceramah Demonstrasi Praktik	Mhs memprogram dengan code-vision	Mhs bisa menggunakan Code vision untuk memprogram mikrokontroler	Penugasan 1		200'	1,2,3,4
2-3	Pemrograman I-O	Sistem minimum Inialisasi Port I-O mikrokontroler Instruksi W/R port I-O	Ceramah Demonstrasi Problem Based Learning (PBL)	Mhs mempersepsi materi ajar sistem minimum Mhs memprogram I-O	Mhs bisa membuat program I-O	Penugasan 2	5%	400'	1,2,3,4
4-6	Pemrograman Timer-Counter	Inialisasi port timer dan counter mikrokontroler	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar sistem minimum Mhs memprogram timer-counter	Mhs bisa membuat program timer-counter	Penugasan 3	10%	400'	1,2,3,4
7	Pemrograman Interupsi	Inialisasi port Interupsi Instruksi Interupsi	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar Interupsi Mhs memprogram Interupsi	Mhs bisa membuat program interupsi	Penugasan 4	5%	200'	1,2,3,4
8	Ujian praktik mid semester					Penugasan 5	20%	200'	
9-10	Aplikasi 7 Segment	Rangkaian 7 segment Pemrograman scanning	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar 7 segment Mhs memprogram 7 segment	Mhs bisa membuat program aplikasi 7 segment	Penugasan 6	10%	200'	1,2,3,4
11	Aplikasi LCD	Instruksi LCD	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar aplikasi LCD Mhs memprogram LCD	Mhs bisa membuat program aplikasi LCD	Penugasan 7	10 %	200'	1,2,3,4
12	Aplikasi Keypad	Pemrograman Scanning	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar keypad	Mhs bisa membuat program aplikasi Keypad	Penugasan 8	20 %	200'	1,2,3,4
13	Aplikasi PWM Motor DC	Pengaturan kecepatan motor dc PWM	Ceramah Demonstrasi PBL	Mhs mempersepsi materi ajar PWM motor DC Mhs memprogram PWM motor dc	Mhs bisa membuat program aplikasi PWM motor DC				

Dibuat oleh: Herlambang SP	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik, Universitas Negeri Yogyakarta	Ketua Prodi :	Dipenaksialeh:
----------------------------	---	---------------	----------------

14.16	Ujian Praktik end semester dan Remedial						600	
-------	--	--	--	--	--	--	-----	--

IV. BOBOT PENILAIAN¹⁾

NO	ASPEK	JENIS TAGIHAN	NILAI MAKSIMAL	BOBOT
1	Kemampuan kognitif & Afektif	Semua tagihan diberi skor (0-100) x bobot tagihan (kolom 8)	Nilai berdasarkan akumulasi capaian skor setiap tagihan	30 %
		UTS ⁷⁾	0-100	25 %
2	Kehadiran	UAS ⁸⁾	0-100	35 %
		Hadir 100 %	100	10 %
		Tidak hadir satu kali	90	
		Tidak hadir dua kali	80	
		Tidak hadir tiga kali	70	
		Tidak hadir empat kali	60	

¹⁾ Penilaian aspek, jenis penilaian dan pembobotan disesuaikan dengan capaian pembelajaran dan karakteristik matkul/ujah

V. SUMBER BACAAAN

1. Ayala, K.J. 1991, *The 8051 Microcontroller Architecture, Programming and Applications*. New York : West Publishing Company.
2. Herlambang Sigit P., 2012, Labsheet Praktikum Mikrokontroler
3. Lingga Wardana, 2006, *Belajar Sendiri Mikrokontroler AVR Seri ATMega 8535*. Yogyakarta: Andi.
4. Rachmad Setiawan, 2006, *Mikrokontroler MCS51*, Graha Ilmu

Dibuat oleh: Herlambang SP	Dilarang memperbarui sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik, Universitas Negeri Yogyakarta	Ketua Prodi :	Diperiksa oleh:
----------------------------	---	---------------	-----------------

Lampiran 2. Foto Produk



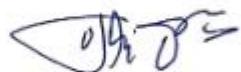
Lampiran 3. Validasi Instrumen

Hasil Validasi Instrumen Penelitian TAS

<p>Nama : Isnan Nabawi NIM : 15518241002 Judul : Pengembangan Sistem Pengaturan Pintu Berbasis <i>Radio Frequency Identification</i> (RFID) Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler</p>		
No	Aspek	Saran/Tanggapan
1	Kisi-kisi	<p>①. Perbaiki kisi-kisi dan teori (hal 9-11). Karena kurang <i>comprehensif</i></p> <p>②. Perbaiki kalimat seperti contoh.</p>
	Komentar umum/lain-lain:	<p>Seorang Umat Kristen, tetapi per konsekuensi yg lebih sesuai dg teori</p>

Yogyakarta,

Validator,



Dr. Edy Supriyadi, M.Pd.

NIP. 19611003 198703 1 002

Scanned by CamScanner

SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Dr. Edy Supriyadi, M.Pd.
NIP : 19611003 198703 1 002
Jurusan : Pendidikan Teknik Elektro
menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Isnan Nabawi
NIM : 15518241002
Program Studi : Pendidikan Teknik Mekatronika
Judul : Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan:

- Layak digunakan untuk penelitian
 Layak digunakan dengan perbaikan
 Tidak layak digunakan untuk penelitian yang bersangkutan

dengan catatan dan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta,

Validator,



Dr. Edy Supriyadi, M.Pd.

NIP. 19611003 198703 1 002

Catatan:

- Beri tanda centang (✓)

Scanned by CamScanner

SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Dr. Samsul Hadi, M.Pd., M.T.
NIP : 19600529 198403 1 003
Jurusan : Pendidikan Teknik Elektro
menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Isnan Nabawi
NIM : 15518241002
Program Studi : Pendidikan Teknik Mekatronika
Judul : Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan:

- Layak digunakan untuk penelitian
 Layak digunakan dengan perbaikan
 Tidak layak digunakan untuk penelitian yang bersangkutan

dengan catatan dan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta,

Validator,



Dr. Samsul Hadi, M.Pd., M.T.

NIP. 19600529 198403 1 003

Catatan:

- Beri tanda centang (✓)

Hasil Validasi Instrumen Penelitian TAS

Nama : Isnain Nabawi
NIM : 15518241002
Judul : Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

No	Aspek	Saran/Tanggapan
1	angket pengguna	a. Perbaiki petunjuk pengguna b. Perbaiki pengetahuan pengguna melalui produk
2.	kin black box	Grafik pola pengeluaran juga supaya dipahami
		Komentar umum/lain-lain:

Yogyakarta, 15/4/2019

Validator,



Dr. Samsul Hadi, M.Pd., M.T.

NIP. 19600529 198403 1 003

Scanned by CamScanner

Lampiran 4. Validasi Ahli Media



ANGKET

(Instrumen Penilaian Media)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : Dr. Syaiful Rizki, M.T.
Institusi/Lembaga : UIN Yogyakarta

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2019

Scanned by CamScanner

ANGKET PENILAIAN AHLI MEDIA

Dalam rangka penelitian tugas akhir skripsi, Saya mohon bantuan Bapak/Ibu/Saudara/i untuk menjadi validator "Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler" agar layak digunakan sebagai media pembelajaran.

Mata Kuliah : Praktik Mikrokontroler

Pembuat : Isnan Nabawi

Tanggal :

Prosedur Pengisian Instrumen Media Pembelajaran:

1. Bapak/Ibu/Saudara/i dimohon untuk memberikan penilaian pada salah satu kolom di dalam angket ini berdasarkan kriteria berikut:

SS : Sangat Setuju

S : Setuju

KS : Kurang Setuju

TS : Tidak Setuju

2. Memberikan tanda centang (✓) pada salah satu pilihan dalam setiap pernyataan sesuai pendapat Bapak/Ibu/Saudara/i.
3. Jika terdapat kekurangan pada media pembelajaran ini, Bapak/Ibu/Saudara/i dapat memberikan saran dan masukan pada tempat yang sudah disediakan.
4. Menandai kesimpulan dengan melingkari nomor yang sesuai dengan pendapat Bapak/Ibu/Saudara/i.
5. Bapak/Ibu/Saudara/i dimohon untuk menuliskan nama sebagai validator serta membubuhkan tanda tangan pada akhir lembar instrumen ini sebagai bukti bahwa instrumen ini dibuat sebenar-benarnya tanpa manipulasi data.

Atas bantuan Bapak/Ibu/Saudara/i, Saya mengucapkan terima kasih.

Scanned by CamScanner

Pernyataan:

No	Pernyataan	Pilihan Jawaban			
		TS	KS	S	SS
1.	Media pembelajaran sesuai dengan materi yang diajarkan	✓			✓
2.	Media pembelajaran sesuai dengan tujuan pembelajaran yang tercantum dalam silabus		✓	✓	
3.	Penggunaan media pembelajaran Sistem Pengaturan Parkir mempermudah proses pembelajaran	✓			✓
4.	Penggunaan media pembelajaran mempermudah peserta didik dalam mempelajari aplikasi dalam mikrokontroler	✓			✓
5.	Media pembelajaran Sistem Pengaturan Parkir mendukung mata kuliah lain yang juga menggunakan mikrokontroler.				✓
6.	Media pembelajaran Sistem Pengaturan Parkir berkaitan dengan materi mata kuliah lain yang juga menggunakan mikrokontroler.				✓
7.	Media pembelajaran Sistem Pengaturan Parkir membantu pengajar dalam menjelaskan materi.				✓
8.	Sistem pengaturan parkir dapat dioperasikan dengan mudah				✓
9.	Proses penghubungan perangkat lunak dengan perangkat keras dapat dilakukan dengan mudah				✓
10.	Tombol pada perangkat keras Sistem Pengaturan Parkir dapat dioperasikan dengan mudah.				✓
11.	Media pembelajaran memiliki bentuk yang menarik				✓
12.	Media pembelajaran memiliki tata letak komponen yang menarik				✓

13.	Media pembelajaran memiliki konstruksi yang kokoh	✓	✓	
14.	Media pembelajaran memiliki ketahanan fisik yang baik untuk proses pembelajaran	✓	✓	
15.	Ukuran media pembelajaran sesuai dengan kebutuhan praktikum.	✓		✓
16.	Media pembelajaran memiliki bentuk yang sesuai dengan kebutuhan praktikum.	✓		✓
17.	Susunan antar bagian sistem pengaturan parkir sudah tersusun rapi.	✓		✓
18.	Informasi pada tampilan LCD mudah dipahami	✓		✓
19.	Warna pada tulisan keterangan dan warna latar belakang sudah sesuai	✓		✓
20.	Penghubungan antar bagian sistem pengaturan parkir dapat dilakukan dengan mudah	✓		✓

Kesimpulan:

Menurut saya, Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler dinyatakan:

1. Layak digunakan tanpa revisi
2. Layak digunakan dengan revisi pada saran
3. Tidak layak digunakan

Saran dan Perbaikan:

label untuk portal masuk/keluar
saklar on/off

Yogyakarta, 15/5/2019
Validator

Deny Budi H
(.....)



ANGKET
(Instrumen Penilaian Media)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : Putri A.
Institusi/Lembaga : UNY

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2019

Scanned by CamScanner

ANGKET PENILAIAN AHLI MEDIA

Dalam rangka penelitian tugas akhir skripsi, Saya mohon bantuan Bapak/Ibu/Saudara/i untuk menjadi validator "Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler" agar layak digunakan sebagai media pembelajaran.

Mata Kuliah : Praktik Mikrokontroler

Pembuat : Isnan Nabawi

Tanggal :

Prosedur Pengisian Instrumen Media Pembelajaran:

1. Bapak/Ibu/Saudara/i dimohon untuk memberikan penilaian pada salah satu kolom di dalam angket ini berdasarkan kriteria berikut:

SS : Sangat Setuju

S : Setuju

KS : Kurang Setuju

TS : Tidak Setuju

2. Memberikan tanda centang (✓) pada salah satu pilihan dalam setiap pernyataan sesuai pendapat Bapak/Ibu/Saudara/i.
3. Jika terdapat kekurangan pada media pembelajaran ini, Bapak/Ibu/Saudara/i dapat memberikan saran dan masukan pada tempat yang sudah disediakan.
4. Menandai kesimpulan dengan melingkari nomor yang sesuai dengan pendapat Bapak/Ibu/Saudara/i.
5. Bapak/Ibu/Saudara/i dimohon untuk menuliskan nama sebagai validator serta membubuhkan tanda tangan pada akhir lembar instrumen ini sebagai bukti bahwa instrumen ini dibuat sebenar-benarnya tanpa manipulasi data.

Atas bantuan Bapak/Ibu/Saudara/i, Saya mengucapkan terima kasih.

Scanned by CamScanner

Pernyataan:

No	Pernyataan	Pilihan Jawaban			
		TS	KS	S	SS
1.	Media pembelajaran sesuai dengan materi yang diajarkan			✓	
2.	Media pembelajaran sesuai dengan tujuan pembelajaran yang tercantum dalam silabus			✓	
3.	Penggunaan media pembelajaran Sistem Pengaturan Parkir mempermudah proses pembelajaran			✓	
4.	Penggunaan media pembelajaran mempermudah peserta didik dalam mempelajari aplikasi dalam mikrokontroler			✓	
5.	Media pembelajaran Sistem Pengaturan Parkir mendukung mata kuliah lain yang juga menggunakan mikrokontroler.		✓		
6.	Media pembelajaran Sistem Pengaturan Parkir berkaitan dengan materi mata kuliah lain yang juga menggunakan mikrokontroler.			✓	
7.	Media pembelajaran Sistem Pengaturan Parkir membantu pengajar dalam menjelaskan materi.			✓	
8.	Sistem pengaturan parkir dapat dioperasikan dengan mudah				✓
9.	Proses penghubungan perangkat lunak dengan perangkat keras dapat dilakukan dengan mudah			✓	
10.	Tombol pada perangkat keras Sistem Pengaturan Parkir dapat dioperasikan dengan mudah.			✓	
11.	Media pembelajaran memiliki bentuk yang menarik			✓	
12.	Media pembelajaran memiliki tata letak komponen yang menarik			✓	

13.	Media pembelajaran memiliki konstruksi yang kokoh		✓	
14.	Media pembelajaran memiliki ketahanan fisik yang baik untuk proses pembelajaran		✓	
15.	Ukuran media pembelajaran sesuai dengan kebutuhan praktikum.		✗✓	
16.	Media pembelajaran memiliki bentuk yang sesuai dengan kebutuhan praktikum.		✓	
17.	Susunan antar bagian sistem pengaturan parkir sudah tersusun rapi.		✓	
18.	Informasi pada tampilan LCD mudah dipahami			✓
19.	Warna pada tulisan keterangan dan warna latar belakang sudah sesuai		✓	
20.	Penghubungan antar bagian sistem pengaturan parkir dapat dilakukan dengan mudah		✓	

Kesimpulan:

Menurut saya, Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler dinyatakan:

1. Layak digunakan tanpa revisi
2. Layak digunakan dengan revisi pada saran
3. Tidak layak digunakan

Saran dan Perbaikan:

- *Caranya portabel & bisa dg tapping RFID card, maka sentar sebutknya d'puluhan setelah parkir / via memanfaatkan teknologi nfc. Atau opsi portabel.*
- *Karena ini masih pembelajaran praktik, mk latihan harus menggunakan modul hardware yg sifatnya 2d buat*
- Yogyakarta, 2 Mei 2019
Validator
- (Lestam / Lestam)

Lampiran 5. Validasi Ahli Materi



ANGKET
(Instrumen Penilaian Materi)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : Syaiful
Institusi/Lembaga : UNY

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2019

Scanned by CamScanner

ANGKET PENILAIAN AHLI MATERI

Dalam rangka penelitian tugas akhir skripsi, Saya mohon bantuan Bapak/Ibu/Saudara/i untuk menjadi validator "Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler" agar layak digunakan sebagai media pembelajaran.

Mata Kuliah : Praktik Mikrokontroler
Pembuat : Isnaw Nabawi
Tanggal :

Prosedur Pengisian Instrumen Materi Pembelajaran:

1. Bapak/Ibu/Saudara/i dimohon untuk memberikan penilaian pada salah satu kolom di dalam angket ini berdasarkan kriteria berikut:

SS : Sangat Setuju
S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju

2. Memberikan tanda centang (✓) pada salah satu pilihan dalam setiap pernyataan sesuai pendapat Bapak/Ibu/Saudara/i.
3. Jika terdapat kekurangan pada media pembelajaran ini, Bapak/Ibu/Saudara/i dapat memberikan saran dan masukan pada tempat yang sudah disediakan.
4. Menandai kesimpulan dengan melingkari nomor yang sesuai dengan pendapat Bapak/Ibu/Saudara/i.
5. Bapak/Ibu/Saudara/i dimohon untuk menuliskan nama sebagai validator serta membubuhkan tanda tangan pada akhir lembar instrumen ini sebagai bukti bahwa instrumen ini dibuat sebenar-benarnya tanpa manipulasi data.

Atas bantuan Bapak/Ibu/Saudara/i, Saya mengucapkan terima kasih.

Scanned by CamScanner

Pernyataan:

No	Pernyataan	Pilihan Jawaban			
		TS	KS	S	SS
1.	Materi pembelajaran yang diajarkan menggunakan media pembelajaran telah sesuai dengan rencana pembelajaran semester (RPS)			✓	
2.	Materi pada <i>jobsheet</i> relevan dengan pembelajaran praktik mikrokontroler			✓	
3.	Penjelasan tentang RFID disajikan lengkap sesuai dengan kebutuhan materi pada mata kuliah praktik mikrokontroler.				✓
4.	Materi pada <i>jobsheet</i> sudah lengkap sesuai dengan kebutuhan praktik.			✓	
5.	Materi tentang RFID pada <i>jobsheet</i> mudah dipahami.			✓	
6.	Materi untuk komponen pendukung sistem pengaturan parkir pada modul mudah dipahami.			✓	
7.	Materi tentang RFID pada <i>jobsheet</i> mudah dipahami.			✓	
8.	Penjelasan tentang tiap komponen yang digunakan dalam sistem pengaturan parkir dijelaskan di dalam modul dengan baik.				✓
9.	Penjelasan tentang alur kerja sistem pengaturan parkir dijelaskan di dalam modul dengan baik.			✓	
10.	Materi tentang pengoperasian sistem pengaturan parkir disajikan dengan jelas		✓		
11.	Materi yang diajarkan dalam <i>jobsheet</i> dan modul meningkatkan pengetahuan siswa			✓	
12.	Sistem pengaturan parkir dapat mendukung materi pada mata kuliah praktik mikrokontroler			✓	

13.	Materi yang disajikan sudah runut sesuai dengan kebutuhan praktik.				✓
14.	Terdapat penjelasan/keterangan pada setiap fungsi yang digunakan dalam program				✓
15.	Materi yang disajikan dapat dipahami dengan mudah			✓	
16.	Materi yang disajikan sesuai dengan perkembangan emosional peserta didik			✓	
17.	Tata bahasa <i>jobsheet</i> sesuai dengan kaidah Bahasa Indonesia				✓
18.	Istilah-istilah pada <i>jobsheet</i> sudah baku.			✓	

Kesimpulan:

Menurut saya, materi pada Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler dinyatakan:

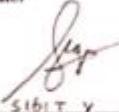
1. Layak digunakan tanpa revisi
2. Layak digunakan dengan revisi pada saran
3. Tidak layak digunakan

Saran dan Perbaikan:

- * *Jobsheet* yg menggunakan portasi dan penanda LCD yg ada pada model parkir dibuat
- * *Jobsheet* tgj RFID pada sistem mekanisme pengaturan harus valid dan benar

Yogyakarta.....

Validator


(.....Sigit Y.....)



ANGKET
(Instrumen Penilaian Materi)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : Agus ...
Institusi/Lembaga : UNY

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2019

Scanned by CamScanner

ANGKET PENILAIAN AHLI MATERI

Dalam rangka penelitian tugas akhir skripsi, Saya mohon bantuan Bapak/Ibu/Saudara/i untuk menjadi validator "Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification (RFID)* Sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler" agar layak digunakan sebagai media pembelajaran.

Mata Kuliah : Praktik Mikrokontroler
Pembuat : Isnan Nabawi
Tanggal :

Prosedur Pengisian Instrumen Materi Pembelajaran:

1. Bapak/Ibu/Saudara/i dimohon untuk memberikan penilaian pada salah satu kolom di dalam angket ini berdasarkan kriteria berikut:
SS : Sangat Setuju
S : Setuju
KS : Kurang Setuju
TS : Tidak Setuju
2. Memberikan tanda centang (✓) pada salah satu pilihan dalam setiap pernyataan sesuai pendapat Bapak/Ibu/Saudara/i.
3. Jika terdapat kekurangan pada media pembelajaran ini, Bapak/Ibu/Saudara/i dapat memberikan saran dan masukan pada tempat yang sudah disediakan.
4. Menandai kesimpulan dengan melengkapi nomor yang sesuai dengan pendapat Bapak/Ibu/Saudara/i.
5. Bapak/Ibu/Saudara/i dimohon untuk menuliskan nama sebagai validator serta membubuhkan tanda tangan pada akhir lembar instrumen ini sebagai bukti bahwa instrumen ini dibuat sebenar-benarnya tanpa manipulasi data.

Atas bantuan Bapak/Ibu/Saudara/i, Saya mengucapkan terima kasih.

Scanned by CamScanner

Pernyataan:

No	Pernyataan	Pilihan Jawaban			
		TS	KS	S	SS
1.	Materi pembelajaran yang diajarkan menggunakan media pembelajaran telah sesuai dengan rencana pembelajaran semester (RPS)				✓
2.	Materi pada <i>jobsheet</i> relevan dengan pembelajaran praktik mikrokontroler				✓
3.	Penjelasan tentang RFID disajikan lengkap sesuai dengan kebutuhan materi pada mata kuliah praktik mikrokontroler.			✓	
4.	Materi pada <i>jobsheet</i> sudah lengkap sesuai dengan kebutuhan praktik.				✓
5.	Materi tentang RFID pada <i>jobsheet</i> mudah dipahami.				✓
6.	Materi untuk komponen pendukung sistem pengaturan parkir pada modul mudah dipahami.			✓	
7.	Materi tentang RFID pada <i>jobsheet</i> mudah dipahami.			✓	
8.	Penjelasan tentang tiap komponen yang digunakan dalam sistem pengaturan parkir dijelaskan di dalam modul dengan baik.				✓
9.	Penjelasan tentang alur kerja sistem pengaturan parkir dijelaskan di dalam modul dengan baik				✓
10.	Materi tentang pengoperasian sistem pengaturan parkir disajikan dengan jelas				✓
11.	Materi yang diajarkan dalam <i>jobsheet</i> dan modul meningkatkan pengetahuan siswa				✓
12.	Sistem pengaturan parkir dapat mendukung materi pada mata kuliah praktik mikrokontroler			✓	

13.	Materi yang disajikan sudah runut sesuai dengan kebutuhan praktik.			✓
14.	Terdapat penjelasan/keterangan pada setiap fungsi yang digunakan dalam program			✓
15.	Materi yang disajikan dapat dipahami dengan mudah			✓
16.	Materi yang disajikan sesuai dengan perkembangan emosional peserta didik		✓	
17.	Tata bahasa <i>jobsheet</i> sesuai dengan kaidah Bahasa Indonesia		✓	
18.	Istilah-istilah pada <i>jobsheet</i> sudah baku.			✓

Kesimpulan:

Menurut saya, materi pada Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler dinyatakan:

1. Layak digunakan tanpa revisi
2. Layak digunakan dengan revisi pada saran
3. Tidak layak digunakan

Saran dan Perbaikan:

- Tata letak tulisan dan kode program pd. *jobsheet* dapat dirapikan

Yogyakarta, 8 Mei 2019

Validator


(Arifin C. N.)

Lampiran 6. Penilaian Pengguna



ANGKET

(Instrumen Penilaian Pengguna)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : HARISTYAH MIFTAH M.
Institusi/Lembaga : UNIVERSITAS NEGERI YOGYAKARTA

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2019

Scanned by CamScanner

B. Angket Penilaian:

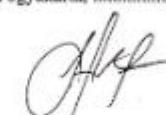
No	Pernyataan	Pilihan Jawaban			
		1	2	3	4
1.	Kalimat yang ada dalam <i>jobsheet</i> mudah dipahami	1	2	✓	4
2.	Materi yang ada dalam <i>jobsheet</i> mudah dipahami	1	2	3	✓
3.	Contoh program yang ada pada <i>jobsheet</i> mudah dipahami.	1	2	✓	4
4.	Materi yang disajikan sesuai dengan mata kuliah praktik mikrokontroler	1	2	✓	4
5.	Materi yang disajikan berisi kompetensi yang saya butuh kan.	1	2	✓	4
6.	Ilustrasi pada <i>jobsheet</i> memudahkan saya belajar.	1	2	3	✓
7.	Langkah kerja pada <i>jobsheet</i> mudah saya ikuti	1	2	3	✓
8.	Bagian-bagian sistem pengaturan parkir tidak membingungkan.	1	2	✓	4
9.	Sistem pengaturan parkir dapat dengan mudah dioperasikan	1	2	✓	4
10.	Modul RFID pada sistem pengaturan parkir dapat saya operasikan.	1	2	3	✓
11.	Modul <i>Real Time Clock</i> pada sistem pengaturan parkir dapat saya operasikan.	1	2	✓	4
12.	Modul SDCard pada sistem pengaturan parkir dapat saya operasikan.	1	2	✓	4
13.	Sistem pengaturan parkir membantu pengajar dalam menjelaskan materi tentang mikrokontroler	1	2	✓	4
14.	Sistem pengaturan parkir menambah kompetensi saya dalam memahami sistem mikrokontroler	1	2	✓	4
15.	Sistem pengaturan parkir memberi saya tambahan pengetahuan dalam pemrograman mikrokontroler	1	2	✓	4

16.	Sistem pengaturan parkir memberi saya tambahan pengetahuan tentang macam-macam aplikasi mikrokontroler.	1	2	3	4
17.	Media pembelajaran ini dapat membantu saya memahami materi pada mata kuliah lain.	1	2	3	4
18.	Sistem pengaturan parkir membantu saya mempelajari mikrokontroler dan pemrogramannya	1	2	3	4
19.	Sistem pengaturan parkir meningkatkan keaktifan saya saat mata kuliah praktik mikrokontroler.	1	2	3	4
20.	Sistem pengaturan parkir menambah motivasi belajar saya saat mata kuliah praktik mikrokontroler.	1	2	3	4

C. Saran dan komentar

Sudah cukup bagus sebagai Media belajar.

Yogyakarta,



(HABASTYA MIFTAH M.)



ANGKET
(Instrumen Penilaian Pengguna)

Pengembangan Sistem Pengaturan Parkir Berbasis *Radio Frequency Identification* (RFID) sebagai Media Pembelajaran Mata Kuliah Praktik Mikrokontroler

IDENTITAS RESPONDEN

Nama : Adenaz Dewandoga
Institusi/Lembaga : UNY

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2019

Scanned by CamScanner

B. Angket Penilaian:

No	Pernyataan	Pilihan Jawaban		
1.	Kalimat yang ada dalam <i>jobsheet</i> mudah dipahami	1	2	3 <input checked="" type="checkbox"/> 4
2.	Materi yang ada dalam <i>jobsheet</i> mudah dipahami	1	2	3 <input checked="" type="checkbox"/> 4
3.	Contoh program yang ada pada <i>jobsheet</i> mudah dipahami.	1	2	3 <input checked="" type="checkbox"/> 4
4.	Materi yang disajikan sesuai dengan mata kuliah praktik mikrokontroler	1	2	3 <input checked="" type="checkbox"/> 4
5.	Materi yang disajikan berisi kompetensi yang saya butuh kan.	1	2	3 <input checked="" type="checkbox"/> 4
6.	Ilustrasi pada <i>jobsheet</i> memudahkan saya belajar.	1	2	3 <input checked="" type="checkbox"/> 4
7.	Langkah kerja pada <i>jobsheet</i> mudah saya ikuti	1	2	3 <input checked="" type="checkbox"/> 4
8.	Bagian-bagian sistem pengaturan parkir tidak membingungkan.	1	2	3 <input checked="" type="checkbox"/> 4
9.	Sistem pengaturan parkir dapat dengan mudah dioperasikan	1	2	3 <input checked="" type="checkbox"/> 4
10.	Modul RFID pada sistem pengaturan parkir dapat saya operasikan.	1	2	3 <input checked="" type="checkbox"/> 4
11.	Modul <i>Real Time Clock</i> pada sistem pengaturan parkir dapat saya operasikan.	1	2	3 <input checked="" type="checkbox"/> 4
12.	Modul SDCard pada sistem pengaturan parkir dapat saya operasikan.	1	2	3 <input checked="" type="checkbox"/> 4
13.	Sistem pengaturan parkir membantu pengajar dalam menjelaskan materi tentang mikrokontroler	1	2	3 <input checked="" type="checkbox"/> 4
14.	Sistem pengaturan parkir menambah kompetensi saya dalam memahami sistem mikrokontroler	1	2	3 <input checked="" type="checkbox"/> 4
15.	Sistem pengaturan parkir memberi saya tambahan pengetahuan dalam pemrograman mikrokontroler	1	2	3 <input checked="" type="checkbox"/> 4

16.	Sistem pengaturan parkir memberi saya tambahan pengetahuan tentang macam-macam aplikasi mikrokontroler.	1	2	3	<input checked="" type="checkbox"/>
17.	Media pembelajaran ini dapat membantu saya memahami materi pada mata kuliah lain.	1	2	3	<input checked="" type="checkbox"/>
18.	Sistem pengaturan parkir membantu saya mempelajari mikrokontroler dan pemrogramannya	1	2	3	<input checked="" type="checkbox"/>
19.	Sistem pengaturan parkir meningkatkan keaktifan saya saat mata kuliah praktik mikrokontroler.	1	2	3	<input checked="" type="checkbox"/>
20	Sistem pengaturan parkir menambah motivasi belajar saya saat mata kuliah praktik mikrokontroler.	1	2	3	<input checked="" type="checkbox"/>

C. Saran dan komentar

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Yogyakarta,



(Aditya Dewangga)

Analisis Data Ahli Materi

Lampiran 7. Analisis Data

No	Validator	Kesesuaian media pembelajaran dengan materi												Aspek						Jml %	Jml %	Jml %						
		1	2	3	4	5	6	7	8	9	10	11	12	Jml %	Penyajian %	Jml %	Bahasa %	Jml %	Jml %									
1	Sigit Yatmono	3	3	4	3	3	3	3	4	3	2	3	3	37,0	L	4	4	8,0	SL	3	4	3	13,0	L	58,0	L		
2	Ariadie Chandra	4	4	3	4	4	3	3	4	4	4	4	3	44,0	SL	4	4	8,0	SL	4	3	3	14,0	SL	66,0	SL		
	Jumlah													81,0				16,0						27,0	124,0			
	Rata-Rata													40,5	L			8,0	SL						13,5	L	62,0	SL
	Presentase													84,4	%			100,0	%						84,4	%	86,1	%
	Jumlah Butir													12				2							4		18,0	
	Skor Maks													48				8							16		72,0	
	Skor Min													12				2							4		18,0	
	Rerata Ideal													30				5							10		45,0	
	Simpangan Ideal													6				1							2		9,0	

Kategori Penilaian	Interval Aspek Relevansi	Interval Aspek Penyajian	Interval Aspek Bahasa	Keseluruhan				Ket																
				1	2	3	4																	
Sangat Layak	40,8 ≤ X < 40,8	6,8 ≤ X < 6,8	13,6 ≤ X < 13,6	61,2	≤	X	61,2	≤	X	61,2	≤	X	61,2	≤	X	61,2	≤	X	61,2	≤	X	61,2	L	
Layak	33,6 ≤ X < 33,6	5,6 ≤ X < 5,6	11,2 ≤ X < 11,2	50,4	≤	X	13,6	≤	X	50,4	≤	X	11,2	≤	X	39,6	≤	X	50,4	≤	X	50,4	CL	
Cukup Layak	26,4 ≤ X < 26,4	4,4 ≤ X < 4,4	8,8 ≤ X < 8,8	28,8	≤	X	4,4	≤	X	28,8	≤	X	6,4	≤	X	39,6	≤	X	39,6	KL				
Kurang Layak	19,2 ≤ X < 19,2	3,2 ≤ X < 3,2	3,2 ≤ X < 3,2	28,8	≤	X	3,2	≤	X	28,8	≤	X	6,4	≤	X	28,8	≤	X	28,8	SK				
Sangat Kurang																								

Analisis Data Ahli Media

No	Validator	Aspek														Total	%	
		Edukatif				Teknik Pembuatan				Estetika				Jml		%		
1	Rustam Asnawi	1	2	3	4	5	6	7	8	9	10	13	14	11	12	15	19	20
1	Rustam Asnawi	3	3	3	3	12,0	1	2	3	4	3	3	24,0	1	3	4	3	24,0
2	Deny Budi Hertanto	4	3	4	4	15,0	5,1	4	4	4	3	4	30,0	5,1	4	4	4	32,0
	Jumlah					27,0							54,0					50,0
	Rata-Rata					13,5							27,0					28,0
	Persentase					84,4							84,4					88,5
	Jumlah Butir					4							8					8
	Skor Maks					16							32					32
	Skor Min					4							8					8
	Rerata ideal					10							20					20
	Simpangan ideal					2							4					4
																		10,0

Kategori Penilaian	Interval Aspek Edukatif	Interval Aspek Teknik Pembuatan				Interval Aspek Estetika				Total		Ket					
		Sangat Layak		Layak		Cukup Layak		Kurang Layak		Sangat Kurang							
Sangat Layak	13,0	≤	X	<	13,0	22,4	≤	X	<	27,2	≤	68,0	≤	X	<	68	SL
Layak	11,2	≤	X	<	11,2	17,0	≤	X	<	22,4	≤	56,0	≤	X	<	56	L
Cukup Layak	8,8	≤	X	<	8,8	12,8	≤	X	<	17,0	≤	44,0	≤	X	<	44	CL
Kurang Layak	6,4	≤	X	<	6,4	0,4	≤	X	<	12,8	≤	32,0	≤	X	<	44,0	KL
Sangat Kurang																	32,0

Analisis Data Uji Pengguna

No	Nama Responden	Aspek																					
		Kualitas Isi & Tujuan					Jml N	Penggunaan						Jml N	Kualitas Pembelajaran								
		1	2	3	4	5		6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	Haristy Miftah M	3	4	3	3	3	16	L	3	3	4	3	3	3	16	L	4	4	3	3	3	3	3
2	Putri Rahmawati	3	3	3	4	3	16	L	4	4	3	3	3	3	17	SL	3	3	4	4	4	4	4
3	Donyansyah Dimas	3	3	4	4	3	17	SL	3	3	4	4	4	4	18	SL	3	4	3	3	3	3	3
4	I Made Yana P	4	4	4	3	3	18	SL	4	4	3	4	3	3	18	SL	4	3	4	4	4	3	4
5	Muhammad Rheza P	3	3	3	3	4	18	L	3	3	4	3	3	3	18	L	4	3	3	3	4	3	3
6	Oriwarda Rio G	4	4	3	3	3	17	SL	4	4	3	3	3	3	17	SL	3	3	3	3	3	3	3
7	Adimas Dewangga	4	4	3	4	4	19	SL	4	3	4	4	4	4	19	SL	4	3	4	4	4	4	4
8	Angela M	3	3	4	3	3	16	L	4	4	4	3	3	3	19	SL	3	4	4	3	3	4	3
9	Inggii Putriana	3	3	3	4	4	17	SL	3	3	3	4	4	4	17	SL	4	4	4	4	4	3	3
10	Martono S	4	4	4	3	4	19	SL	4	3	4	3	4	3	18	SL	3	4	3	4	3	4	4
11	Anis Lestari	4	3	4	4	4	19	SL	4	4	4	3	4	3	19	SL	4	4	3	4	4	4	4
12	Adik Miftah	3	3	3	4	3	16	L	4	4	4	4	3	3	19	SL	3	4	4	4	4	4	4
13	Reva Rahmadibya	3	4	3	3	3	16	L	4	3	3	3	3	3	18	L	3	3	3	4	4	4	4
14	M Garin Akautsar R	3	3	4	3	3	16	L	4	3	4	4	4	4	19	SL	4	4	4	4	4	4	4
15	Aji Purmaputra	3	3	3	4	3	16	L	4	4	2	3	3	3	18	L	2	3	4	3	2	3	4
16	Hermawan Galih	3	4	3	4	3	17	SL	3	3	4	3	3	3	18	L	3	4	3	3	3	3	3
17	Ted Harison	4	3	3	3	3	16	L	3	4	3	3	3	3	18	L	3	3	3	4	3	3	3
18	Kevin Dexsyam	4	4	4	4	4	20	SL	4	4	4	4	3	3	19	SL	4	4	4	4	4	4	4
19	M. Bregas B	3	3	4	4	4	18	SL	4	3	4	3	3	3	17	SL	4	4	3	4	4	4	4
20	Fendi Agus S	3	3	3	3	4	16	L	3	4	4	4	4	4	18	SL	3	3	4	4	4	4	4
		Jumlah					341					351					704					1398	
		Kata-Kata					17,1 SL					17,6 SL					35,2 SL					69,8 SL	
		Presentase					83,3 %					87,8 %					88,0 %					87,3 %	
		Jumlah Butir					5					5					10					20	
		Skor Maks					20					20					40					80	
		Skor Min					5					5					10					20	
		Rerata Ideal					12,5					13					25					50	
		Simpanangan Ideal					2,5					2,5					5					10	

Kategori Penilaian	Interval Aspek		Interval Aspek		Interval Aspek		Keseluruhan	Ket
	Kualitas Isi & Tujuan	Kualitas Penggunaan	Kualitas Penggunaan	Kualitas Pembelajaran	Kualitas Pembelajaran	Kualitas Pembelajaran		
Sangat Layak	17,0 ≤ X	17,0 ≤ X	34,0 ≤ X	34,0 ≤ X	68,0 ≤ X	68,0 ≤ X	SL	
Layak	14,0 ≤ X < 17,0	14,0 ≤ X < 17,0	28,0 ≤ X < 34,0	28,0 ≤ X < 34,0	56,0 ≤ X < 68,0	56,0 ≤ X < 68,0	L	
Cukup Layak	11,0 ≤ X < 14,0	11,0 ≤ X < 14,0	22,0 ≤ X < 28,0	22,0 ≤ X < 28,0	44,0 ≤ X < 56,0	44,0 ≤ X < 56,0	CL	
Kurang	8,0 ≤ X < 11,0	8,0 ≤ X < 11,0	16,0 ≤ X < 22,0	16,0 ≤ X < 22,0	32,0 ≤ X < 44,0	32,0 ≤ X < 44,0	K	
Sangat Kurang	X < 8,0	X < 8,0	X < 16,0	X < 16,0	X < 32,0	X < 32,0	SK	

No.	Nama	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Haristy Miftah M	3	4	3	3	3	4	4	3	3	4	3	3	3	3	3	3	3	3	3	3
2	Putri Rahmawati	3	3	3	4	3	3	3	4	4	3	3	3	3	4	4	4	4	4	4	4
3	Donyansyah Dimas	3	3	4	4	3	3	4	3	3	4	4	4	4	3	3	4	3	3	3	4
4	I Made Yana P	4	4	4	3	3	4	3	4	4	4	3	4	3	4	4	4	3	4	3	4
5	Muhammad Rheza P	3	3	3	3	4	4	3	3	3	4	3	3	3	3	3	4	3	3	3	3
6	Oriwarda Rio G	4	4	3	3	3	3	3	4	4	3	3	3	3	3	3	4	3	3	3	3
7	Adimas Dewangga	4	4	3	4	4	4	3	4	3	4	4	4	4	4	4	4	4	4	4	4
8	Angela M	3	3	4	3	3	3	4	4	4	4	4	4	4	3	3	4	3	3	2	3
9	Inggii Putriana	3	3	3	4	4	4	4	4	3	3	3	4	4	4	4	4	3	3	3	3
10	Martono S	4	4	4	3	4	3	4	4	4	3	4	3	4	3	4	3	4	3	4	4
11	Anis Lestari	4	3	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	3	4	4
12	Adik Miftah	3	3	3	4	3	3	4	4	4	4	4	4	4	3	4	4	4	4	3	4
13	Reva Rahmadibya	3	4	3	3	3	3	4	3	3	3	3	3	3	3	4	4	4	4	4	4
14	M Garin Akautsar R	3	3	4	3	3	4	4	4	3	4	4	4	4	4	4	4	3	4	3	4
15	Aji Purmaputra	3	3	3	4	3	2	3	4	4	2	3	3	4	4	3	3	2	3	4	4
16	Hermawan Galih	3	4	3	4	3	3	4	3	3	3	4	3	3	3	3	3	3	3	4	3
17	Ted Harison	4	3	3	3	3	3	3	3	4	3	3	3	3	3	3	4	3	3	4	3
18	Kevin Dexsyam	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4
19	M. Bregas B	3	3	4	4	4	4	4	4	4	3	4	3	3	3	4	4	4	4	4	4
20	Fendi Agus S	3	3	3	3	4	3	3	3	4	4	4	4	4	4	4	3	4	4	4	4

Lampiran 8. Dokumentasi



Lampiran 9. SK Pembimbing

KEPUTUSAN DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : II/PMEK/PI/II/2019

TENTANG
PENGANGKATAN DOSEN PEMBIMBING TUGAS AKHIR SKRIPSI (TAS) MAHASISWA
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

DEKAN FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

- Menimbang : a. bahwa untuk kelancaran pelaksanaan kegiatan Tugas Akhir Skripsi (TAS) mahasiswa, dipandang perlu mengangkat dosen pembimbingnya;
b. bahwa untuk keperluan sebagaimana dimaksud pada huruf a perlu menetapkan Keputusan Dekan Tentang Pengangkatan Dosen Pembimbing Tugas Akhir Skripsi (TAS) Fakultas Teknik Universitas Negeri Yogyakarta.
- Mengingat : 1. Undang-undang RI Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional (Lembaran Negara Tahun 2003 Nomor 78, Tambahan Lembaran Negara Nomor 4301);
2. Peraturan Pemerintah Republik Indonesia Nomor 4 Tahun 2014 Tentang Penyelenggaraan Pendidikan Tinggi dan Pengelolaan Perguruan Tinggi (Lembaran Negara Tahun 2014 Nomor 16, Tambahan Lembaran Negara Republik Indonesia Nomor 5500);
3. Keputusan Presiden Republik Indonesia Nomor 93 Tahun 1999 Tentang Perubahan Institut Keguruan dan Ilmu Pendidikan menjadi Universitas;
4. Peraturan Mendiknas RI Nomor 23 Tahun 2011 Tentang Organisasi dan Tata Kerja Universitas Negeri Yogyakarta;
5. Peraturan Mendiknas RI Nomor 34 Tahun 2011 Tentang Statuta Universitas Negeri Yogyakarta;
6. Keputusan Menteri Pendidikan dan Kebudayaan RI Nomor 98/MPK.A4/KP/2013 Tentang Pengangkatan Rektor Universitas Negeri Yogyakarta;
7. Peraturan Rektor Nomor 2 Tahun 2014 tentang Peraturan Akademik;
8. Keputusan Rektor Nomor 800/UN.34/KP/2016 tahun 2016 tentang Pengangkatan Dekan Fakultas Teknik Universitas Negeri Yogyakarta.

MEMUTUSKAN

- Menetapkan : KEPUTUSAN DEKAN TENTANG PENGANGKATAN DOSEN PEMBIMBING TUGAS AKHIR SKRIPSI (TAS) FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA.

PERTAMA : Mengangkat Saudara :

Nama : Herlambang Sigit Pramono, S.T., M.Cs
NIP : 19650829 199903 1 001
Pangkat/Golongan : Penata Tk.I, III/d
Jabatan Akademik : Lektor

sebagai Dosen Pembimbing Untuk mahasiswa penyusun Tugas Akhir Skripsi (TAS) :

Nama : Isman Nabawi
NIM : 15518241002
Prodi Studi : Pend. Teknik Mekatronika - S1
Judul Skripsi/TA : Pengembangan Sistem Pengaturan Parkir berbasis RFID sebagai media pembelajaran mata kuliah praktik mikrokontroller.

- KEDUA** : Dosen Pembimbing sebagaimana dimaksud dalam Diktum PERTAMA bertugas merencanakan, mempersiapkan, melaksanakan, dan mempertanggungjawabkan pelaksanaan kegiatan bimbingan terhadap mahasiswa sebagaimana dimaksud dalam Diktum PERTAMA sampai mahasiswa dimaksud dinyatakan lulus.
- KETIGA** : Biaya yang diperlukan dengan adanya Keputusan ini dibebankan pada Anggaran DIPA Universitas Negeri Yogyakarta Tahun 2019.
- KEEMPAT** : Keputusan ini berlaku sejak tanggal 28 Februari 2019.

Tembusan Keputusan Dekan ini disampaikan kepada :

1. Para Wakil Dekan Fakultas Teknik;
 2. Kepala Bagian Tata Usaha Fakultas Teknik;
 3. Kepala Subbagian Keuangan dan Akuntansi Fakultas Teknik;
 4. Kepala Subbagian Pendidikan Fakultas Teknik;
 5. Ketua Jurusan Pendidikan Teknik Elektro Fakultas Teknik;
 6. Mahasiswa yang bersangkutan;
- Universitas Negeri Yogyakarta.

Ditetapkan di : Yogyakarta
Pada tanggal : 28 Februari 2019



Scanned by CamScanner

Lampiran 10. Izin Penelitian

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
UNIVERSITAS NEGERI YOGYAKARTA
FAKULTAS TEKNIK
Alamat : Kampus Kartanegara, Yogyakarta, 55281
Telp. (0274) 586168 psw. 276,289,292 (0274) 586734 Fax. (0274) 586734
Laman: ft.uny.ac.id E-mail: ft@uny.ac.id, teknik@uny.ac.id

Nomor : 233/UN34.15/LT/2019 30 April 2019
Lamp. : 1 Bendel Proposal
Hal : Izin Penelitian

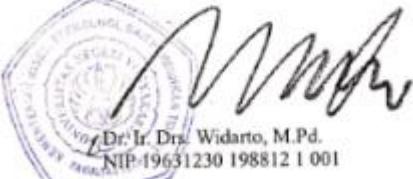
Vth . **Ketua Jurusan Pendidikan Teknik Elektro**
Fakultas Teknik Univeristas Negeri Yogyakarta
di Yogyakarta

Kami sampaikan dengan hormat, bahwa mahasiswa tersebut di bawah ini:

Nama	:	Isnur Nabawi
NIM	:	15518241002
Program Studi	:	Pend. Teknik Mekatronika - S1
Tujuan	:	Memohon izin mencari data untuk penulisan Tugas Akhir Skripsi (TAS)
Judul Tugas Akhir	:	PENGEMBANGAN SISTEM PENGATURAN PARKIR BERBASIS RADIO FREQUENCY IDENTIFICATION (RFID) SEBAGAI MEDIA PEMBELAJARAN MATA KULIAH PRAKTIK MIKROKONTROLER
Waktu Penelitian	:	1 - 31 Mei 2019

Untuk dapat terlaksananya maksud tersebut, kami mohon dengan hormat Bapak/Ibu berkenan memberi izin dan bantuan seperlunya.
Demikian atas perhatian dan kerjasamanya kami sampaikan terima kasih.

Dekan,


Dr. Ir. Drs. Widarto, M.Pd.
NIP.19631230 198812 1 001

Tembusan :
1. Sub. Bagian Pendidikan dan Kemahasiswaan ;
2. Mahasiswa yang bersangkutan.

Scanned by CamScanner

Lampiran 11. Program C# pada Aplikasi simulasi pengisian ID MicroSD

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
namespace SDTEST
{
    public partial class Form1 : Form
    {
        String databaseRead = "";
        UInt32 dbaseLine = 0;
        UInt32 tempFile = 0;
        UInt32 updateID = 0;
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if(folderBrowserDialog1.ShowDialog()==DialogResult.OK)
            {
                label1.Text = folderBrowserDialog1.SelectedPath;
                button2.Enabled = true;
            }
        }
        private void button2_Click(object sender, EventArgs e)
        {
            if(button2.Text=="MULAI")
            {
                button2.Text = "STOP";
                dbaseLine = Convert.ToInt32(textBox4.Text);
                tempFile = Convert.ToInt32(textBox4.Text);
                updateID = Convert.ToInt32(textBox3.Text);
                Directory.CreateDirectory(label1.Text + "DATABASETEST");
                Directory.CreateDirectory(label1.Text + "TEMPORARYTEST");
                StreamWriter sw = new StreamWriter
                    (label1.Text + "DATABASETEST\\DATABASE.txt");
                sw.WriteLine("MASTER 8C59EF49\r\n");
                sw.Close();
                StreamWriter stg = new StreamWriter
                    (label1.Text + "DATABASETEST\\SETTING.txt");
                stg.WriteLine("100A5B6C19D");
                stg.Close();
                timer1.Enabled = true;
            }
            else
            {
                timer1.Enabled = false;
                button2.Text = "MULAI";
            }
        }
        private void timer1_Tick(object sender, EventArgs e)
        {
```

```

tempFile = tempFile + 1;
textBox2.Text = tempFile.ToString();
StreamWriter tmps = new StreamWriter
(label1.Text + "TEMPORARYTEST\\\" + textBox2.Text + ".txt", true);
tmps.Write("1 99");
tmps.Close();
if (tempFile % updateID == 0)
{
    dbaseline = dbaseline + Convert.ToInt16(updateID);
    textBox1.Text = dbaseline.ToString();
    databaseRead = File.ReadAllText
(label1.Text + "DATABASETEST\\DATABASE.txt");
    String idlooping = "";
    for (int i = 0; i < updateID; i++)
    {
        idlooping = idlooping + "MASTER 8C59EF49\r\n";
    }
    StreamWriter dbs = new StreamWriter
(label1.Text + "DATABASETEST\\DATABASE.txt");
    dbs.Write(databaseRead + idlooping);
    dbs.Close();
}
}

private void Form1_Load(object sender, EventArgs e)
{
    updateID = Convert.ToInt32(textBox3.Text);
    textBox1.Text = "0";
    textBox2.Text = "0";
}

private void button3_Click(object sender, EventArgs e)
{
    tempFile = 0;
    dbaseline = 0;
    textBox1.Text = "0";
    textBox2.Text = "0";
    databaseRead = "";
}

private void textBox3_Leave(object sender, EventArgs e)
{
    updateID = Convert.ToInt32(textBox3.Text);
}
}
}

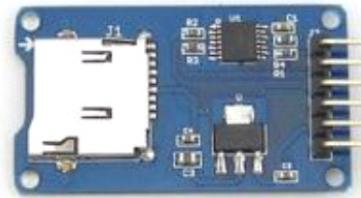
```

Lampiran 12. *Labsheet* Praktikum

Labsheet Praktikum

Praktik Mikrokontroler

- Mengakses Modul RTC DS3231
- Mengakses Modul MicroSD
- Mengakses RFID RC-522
- Aplikasi pada Sistem Pengaturan Parkir



	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:1 dari 9

A. Tujuan

1. Mengetahui konfigurasi RTC DS3231
2. Mengetahui dasar pemrograman RTC DS3231 dengan Arduino
3. Mengetahui cara menampilkan nilai waktu RTC DS3231 pada serial monitor
4. Mengetahui cara menyinkronkan waktu pada RTC DS3231

B. Dasar Teori

1. Arduino Mega

Arduino adalah sebuah perangkat keras elektronik dari sebuah platform yang bersifat *open source* dengan *software* yang mudah digunakan yaitu Arduino *Integrated Development Environment* (IDE). Terdapat berbagai jenis Arduino seperti Arduino nano, Arduino uno, Arduino fiod dan masih banyak lagi. Arduino dapat digunakan dalam berbagai jenis perangkat elektronik yang memerlukan pemrograman dan dapat membaca berbagai jenis sensor sehingga banyak dari kalangan teknisi elektronik menggunakan Arduino untuk melengkapi proyek-proyeknya. Selain itu Arduino juga banyak dimanfaatkan sebagai pembelajaran di kalangan universitas.

Arduino Mega 2560 adalah modul mikrokontroler yang berbasis pada IC ATmega2560. Dia memiliki 54 digital *input/output* (I/O) dengan 14 pin di antaranya mendukung *pulse width modulation* (PWM), 16 *input* analog, 4 jalur *Universal Asynchronous Receiver-Transmitter* (UART), 16 MHz *crystal oscillator*, sebuah koneksi *universal serial bus* (USB), konektor daya, konektor *in circuit serial programming* (ICSP), dan sebuah tombol reset. Dia juga memiliki keseluruhan komponen yang dapat mendukung jalannya mikrokontroler atau sering disebut dengan sistem minimum.



Gambar 1. Arduino Mega
(Sumber: <https://store.arduino.cc/>)

Dibuat oleh: Isnab Nabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
------------------------------	---	-----------------

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
	SEMESTER 4	MENGAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit	
	No. 1	Revisi: 01	Tgl: April 2019	Hal:2 dari 9

2. Real Time Clock (RTC) DS3231

Real-time-clock (RTC) merupakan suatu modul yang memiliki IC (*integrated circuit*) dan baterai yang berfungsi untuk menyimpan informasi waktu dan tanggal secara terus menerus walaupun tidak mendapat pasokan daya dari luar (Suhaeb dkk., 2017:155). Salah satu IC pada modul RTC adalah DS3231. IC DS 3231 menggunakan I2C (*inter integrated circuit*) sebagai antarmuka komunikasinya. RTC memiliki keistimewaan yaitu memiliki sinyal keluaran *programable squarewave*, deteksi kegagalan daya, konsumsi daya kurang dari 500mA, serta menggunakan mode baterai untuk operasional osilator sehingga dapat tetap menyimpan informasi waktu yang akurat terus menerus. RTC memiliki ketahanan suhu dari -40 Celcius hingga +85 Celcius. Adapun fungsi tiap pin dapat dilihat pada Tabel 1.

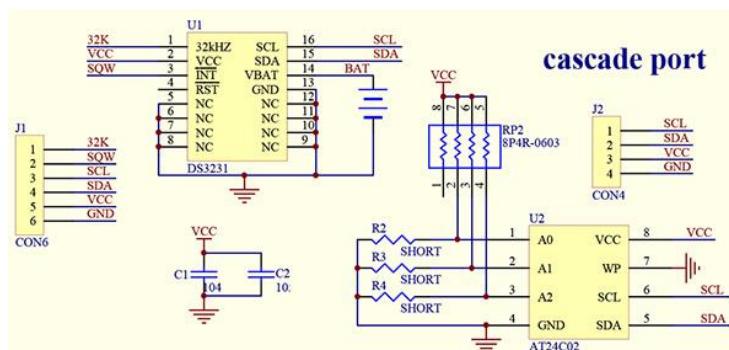
Tabel 1. Fungsi tiap pin modul RTC DS3231

Label pin	Fungsi
VCC	Sumber daya positif 5V
GND	Sumber daya negatif 0V
SDA	<i>Serial Data pin (antarmuka I2C)</i>
SCL	<i>Serial Clock pin (antarmuka I2C)</i>
SQW	<i>Square wave output pin</i>
32K	<i>32K oscillator output.</i>



Gambar 2. Modul Real Time Clock (RTC)

(Sumber: <https://edwardmallon.files.wordpress.com/>)



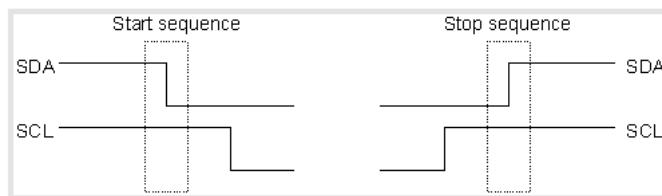
Gambar 3. Skema modul RTC DS3231

(Sumber: <https://edwardmallon.files.wordpress.com/>)

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:3 dari 9

3. Komunikasi I2C

Inter Integrated Circuit (I2C) adalah standar komunikasi serial dua arah yang menggunakan dua saluran yang didesain untuk mengirim maupun menerima data (Suhaeb, dkk., 2017: 154). Komunikasi pada I2C menggunakan 2 pin yaitu *Serial Clock* (SCL) dan *Serial Data* (SDA) yang membawa informasi data antara I2C dengan pengontrolnya. Sinyal *start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 1 menjadi 0 pada saat SCL 1. Sinyal *stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 0 menjadi 1 pada saat SCL 1.



Gambar 4. Start dan Stop bit pada I2C

(Sumber: <https://proyekarduino.wordpress.com/>)

Meski hanya memiliki 2 pin komunikasi, I2C memiliki kemampuan untuk digunakan *multi-slave* karena I2C memiliki sistem pengalamatan. Sehingga jika dimiliki 1 *master* dan 3 *slave*, pin yang digunakan pada master hanya 2 pin (SDA dan SCL) saja. Contoh komponen yang menggunakan komunikasi serial I2C adalah RTC. Pemrograman Komunikasi Serial I2C pada Arduino menggunakan *library "Wire.h"*.

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:4 dari 9

C. Alat dan Bahan

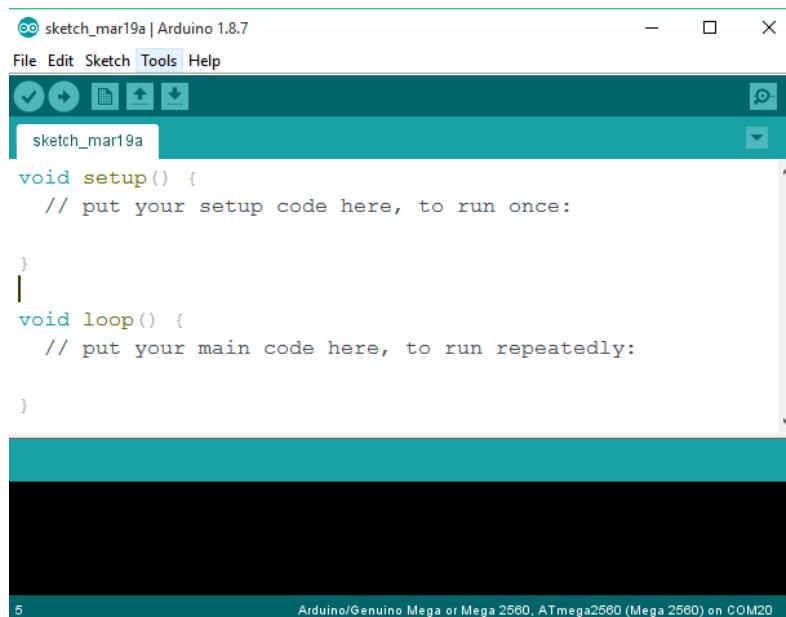
1. Sistem Pengaturan Parkir berbasis RFID (RTC dan Arduino Mega)
2. Kabel USB
3. Komputer dengan *Software Arduino IDE* yang sudah terpasang.

D. Keselamatan Kerja

1. Pastikan saat memulai membuat rangkaian dalam kondisi *off* atau tanpa tegangan.
2. Gunakan alat dan bahan sesuai dengan fungsinya.
3. Jauhkan alat dan bahan yang tidak digunakan dari meja kerja.
4. Dalam merangkai pastikan sesuai dengan langkah kerja yang sudah ditentukan.
5. Tanyalah kepada pengampu apabila terjadi kesalahan dalam pengoperasian dan apabila terdapat penjelasan atau materi yang kurang dimengerti.

E. Langkah Kerja

1. Memasang library RTC DS3231 pada Arduino IDE
 - a. Buka software arduino IDE, maka akan muncul tampilan utama.



```
sketch_mar19a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_mar19a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

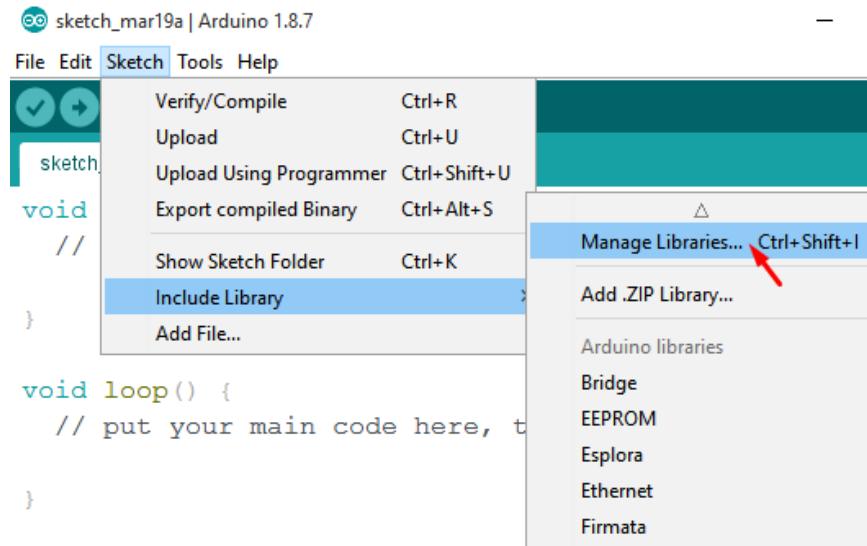
5
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM20
```

Gambar 5. Tampilan utama Arduino IDE

- b. Koneksikan Komputer ke internet kemudian pilih "Sketch" – pilih "Include Library" – pilih "Manage Libraries".

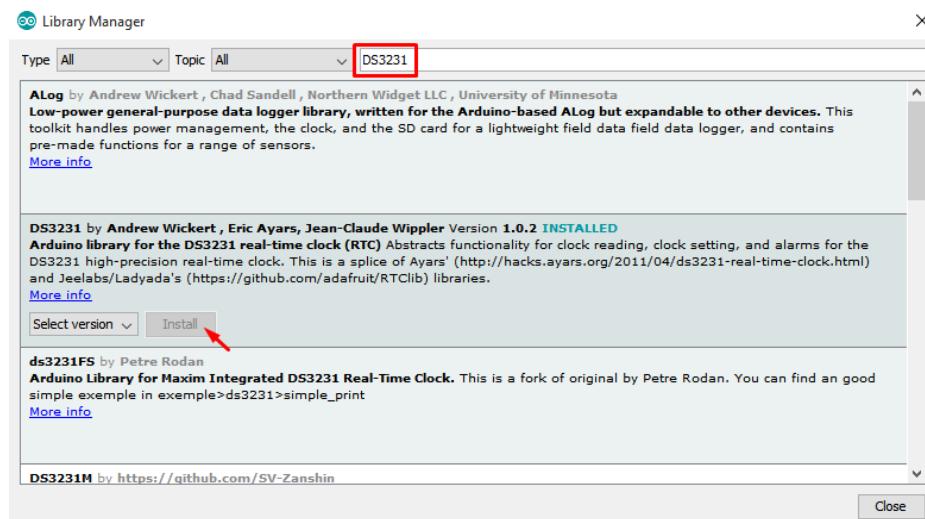
Dibuat oleh: Isnabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
--------------------------	---	-----------------

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:5 dari 9



Gambar 6. Membuka Manage Libraries

Setelah itu akan muncul jendela “Library Manager” – kemudian ketikan “DS3231” dan tekan “Enter” – Klik “Install” – kemudian tunggu proses hingga selesai.

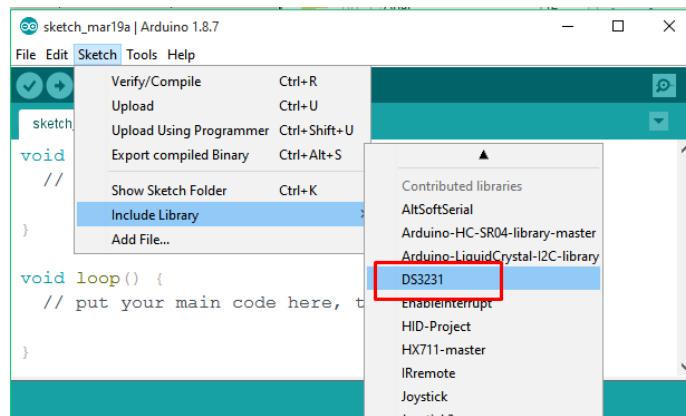


Gambar 7. Tampilan Library Manager

- Untuk menguji apakah library DS3231 sudah masuk atau belum silakan klik “Sketch” – kemudian geser kursor ke “Include Library” – pastikan ada *library* DS3231 di situ.

Dibuat oleh: Isnain Nabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
-------------------------------	---	-----------------

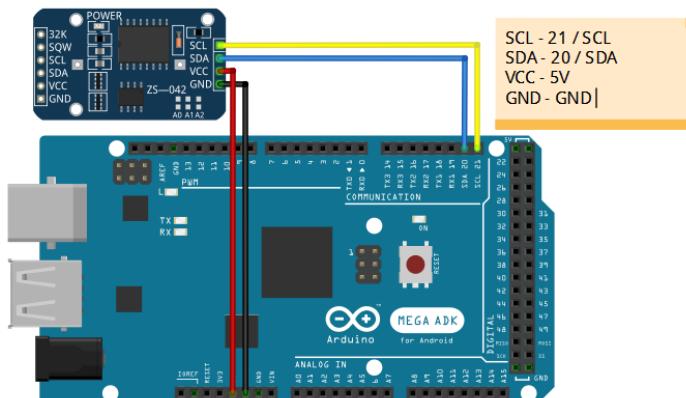
	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019



Gambar 8. Cek *library*

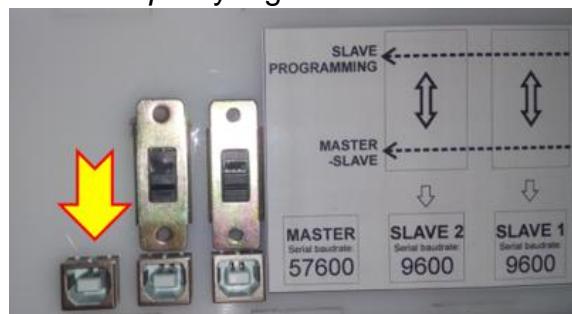
2. Mengidentifikasi Rangkaian Arduino Mega dengan RTC DS3231

Modul RTC dan Arduino Mega sudah tersambung dan terangkai dalam Sistem Pengaturan Parkir dengan koneksi seperti Gambar 9.



Gambar 9. Skematik pengabelan RTC DS3231 dengan Arduino Mega pada Sistem Pengaturan Parkir

3. Menghubungkan Arduino Mega (master) pada Sistem Pengaturan Parkir dengan komputer melalui *port* yang sudah tersedia.



Gambar 10. Konektor USB ke master

Dibuat oleh: Isnain Nabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
-------------------------------	---	-----------------

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENGAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:7 dari 9

4. Memprogram dengan Arduino Ide.

Buka Arduino IDE kemudian isikan program di bawah ini:

```
#include <Wire.h>
#include <DS3231.h>
DS3231 Clock;
bool Century=false;
bool h12;
bool PM;
byte Year;
byte Month;
byte Date;
byte DoW;
byte Hour;
byte Minute;
byte Second;
void GetDateStuff(byte& Year, byte& Month, byte& Day,
                  byte& DoW, byte& Hour, byte& Minute,
                  byte& Second)
{
  // Call this if you notice something coming in on
  // the serial port. The stuff coming in should be in
  // the order YYMMDDwHHMMSS, with an 'x' at the end.
  // YYMMDDwHHMMSS
  // 190213w130900x
  boolean GotString = false;
  char InChar;
  byte Temp1, Temp2;
  char InString[20];
  byte j=0;
  while (!GotString) {
    if (Serial.available()) {
      InChar = Serial.read();
      InString[j] = InChar;
      j += 1;
      if (InChar == 'x') {
        GotString = true;
      }
    }
  }
  Serial.println(InString);
  // Read Year first
  Temp1 = (byte)InString[0] -48;
  Temp2 = (byte)InString[1] -48;
```

Dibuat oleh: Isnab Nabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
------------------------------	---	-----------------

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENGAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:8 dari 9

```

Year = Temp1*10 + Temp2;
// now month
Temp1 = (byte)InString[2] -48;
Temp2 = (byte)InString[3] -48;
Month = Temp1*10 + Temp2;
// now date
Temp1 = (byte)InString[4] -48;
Temp2 = (byte)InString[5] -48;
Day = Temp1*10 + Temp2;
// now Day of Week
Dow = (byte)InString[6] - 48;
// now Hour
Temp1 = (byte)InString[7] -48;
Temp2 = (byte)InString[8] -48;
Hour = Temp1*10 + Temp2;
// now Minute
Temp1 = (byte)InString[9] -48;
Temp2 = (byte)InString[10] -48;
Minute = Temp1*10 + Temp2;
// now Second
Temp1 = (byte)InString[11] -48;
Temp2 = (byte)InString[12] -48;
Second = Temp1*10 + Temp2;
}
void setup()
{
  Wire.begin();
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available()) {
    GetDateStuff(Year, Month, Date, Dow, Hour, Minute, Second);
    Clock.setClockMode(false); // set to 24h
    //setClockMode(true); // set to 12h
    Clock.setYear(Year);
    Clock.setMonth(Month);
    Clock.setDate(Date);
    Clock.setDow(Dow);
    Clock.setHour(Hour);
    Clock.setMinute(Minute);
    Clock.setSecond(Second);
  }
  Serial.print("Jam ");
}

```

Dibuat oleh: Isnab Nabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
------------------------------	---	-----------------

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>REAL TIME CLOCK (RTC) DS3231</i> DEGAN ARDUINO MEGA	4x45 menit
	No. 1	Revisi: 01	Tgl: April 2019 Hal:9 dari 9

```

Serial.print(Clock.getHour(h12, PM));
Serial.print(":");
Serial.print(Clock.getMinute());
Serial.print(":");
Serial.print(Clock.getSecond());

Serial.println("");
Serial.print("Tgl ");
Serial.print(Clock.getDate());
Serial.print("/");
Serial.print(Clock.getMonth(Century));
Serial.print("/");
Serial.print(Clock.getYear());
delay(1000);
}

```

5. *Upload* program ke perangkat.
6. Jalankan serial monitor pada Arduino IDE.
7. Kirim perintah ke Arduino melalui serial monitor kemudian lihat hasilnya dan masukkan pada tabel di bawah ini:

No	Kirim perintah Serial	Output serial monitor
1	(tidak ada perintah)	
2	000000w000000x	
3	ABCDEF	
4	190213w130900x	
5	(isikan waktu saat ini) Format: YYMMDDwHHMMSSx	

F. Bahan Diskusi

Sistem Pengaturan Parkir sudah dilengkapi dengan LCD 16x2 dengan koneksi I2C pada alamat 0x27 dan *keypad* 4x4 yang sudah terhubung dengan Arduino Mega pada pin 30,32,34,36 untuk baris dan pin 22,24,26,28 untuk kolom. Buatlah program untuk menampilkan waktu pada LCD dengan format “Jam:Menit Tgl/Bln/Tahun” dan dapat melakukan sinkronisasi waktu menggunakan *keypad*!

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL <i>READER-WRITER MICROSD</i> DENGAN ARDUINO MEGA	4x45 menit
	No. 02	Revisi: 01	Tgl: April 2019

A. Tujuan

1. Mengetahui konfigurasi Modul MICROSD
2. Mengetahui dasar pemrograman Modul MICROSD dengan Arduino
3. Mengetahui cara menulis data ke Modul MICROSD
4. Mengetahui cara membaca data dari Modul MICROSD

B. Dasar Teori

1. Arduino Mega

Arduino adalah sebuah perangkat keras elektronik dari sebuah platform yang bersifat *open source* dengan *software* yang mudah digunakan yaitu Arduino *Integrated Development Environment* (IDE). Terdapat berbagai jenis Arduino seperti Arduino nano, Arduino uno, Arduino fiod dan masih banyak lagi. Arduino dapat digunakan dalam berbagai jenis perangkat elektronik yang memerlukan pemrograman dan dapat membaca berbagai jenis sensor sehingga banyak dari kalangan teknisi elektronik menggunakan Arduino untuk melengkapi proyek-proyeknya. Selain itu Arduino juga banyak dimanfaatkan sebagai pembelajaran di kalangan universitas.

Arduino Mega 2560 adalah modul mikrokontroler yang berbasis pada IC ATmega2560. Dia memiliki 54 digital *input/output* (I/O) dengan 14 pin di antaranya mendukung *pulse width modulation* (PWM), 16 *input* analog, 4 jalur *Universal Asynchronous Receiver-Transmitter* (UART), 16 MHz *crystal oscillator*, sebuah koneksi *universal serial bus* (USB), konektor daya, konektor *in circuit serial programming* (ICSP), dan sebuah tombol reset. Dia juga memiliki keseluruhan komponen yang dapat mendukung jalannya mikrokontroler atau sering disebut dengan sistem minimum.



Gambar 1. Arduino Mega
(Sumber: <https://store.arduino.cc/>)

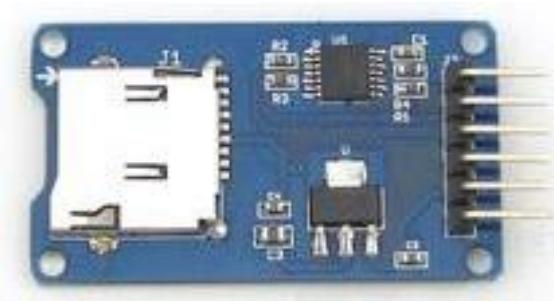
	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL <i>READER-WRITER MICROSD</i> DENGAN ARDUINO MEGA	4x45 menit	
No. 02	Revisi: 01	Tgl: April 2019	Hal: 2 dari 11

2. Modul MicroSD

Modul (MicroSD Card Adapter) adalah modul untuk membaca dan menulis data pada kartu memori mikro yang menggunakan antarmuka SPI (Faudin, 2018). Modul ini cocok digunakan untuk aplikasi yang membutuhkan penyimpanan basis data seperti sistem presensi, antrean, *data logging*, sistem parkir, dan sebagainya. Secara rinci fungsi kaki-kaki pada modul RFID dapat dilihat pada Tabel 1.

Tabel 1. Fungsi kaki-kaki pada modul *micro sd card adapter*

Pin	Label	Fungsi
1	CS	Pin <i>Slave Select</i> untuk komunikasi SPI
2	SCK	Pin <i>Serial Clock</i> untuk komunikasi SPI
3	MOSI	Pin <i>MOSI</i> untuk komunikasi SPI
4	MISO	Pin <i>MISO</i> untuk komunikasi SPI
5	VCC	Sumber daya positif 5V
6	GND	Sumber daya negatif 0V



Gambar 2. Modul SD Card Adapter

(Sumber: <http://www.indo-ware.com>)

3. Komunikasi SPI

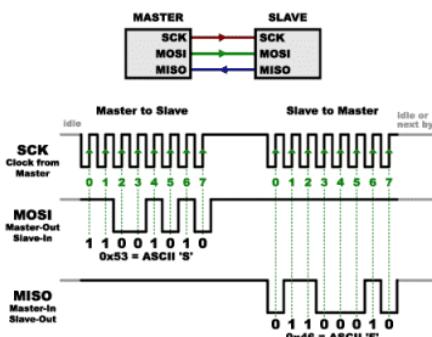
Serial Peripheral Interface (SPI) merupakan komunikasi serial sinkron karena menggunakan jalur terpisah antara data dan *clock* (Suhaeb dkk., 2017:135). Jalur data dan *clock* yang terpisah membuat komunikasi SPI



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4	MENGAKSES MODUL <i>READER-WRITER MICROSD</i> DENGAN ARDUINO MEGA	
No. 02	Revisi: 01	Tgl: April 2019

menjadi lebih cepat karena data pada pengirim dan penerima lebih akurat. Pada umumnya komunikasi menggunakan SPI memiliki beberapa jalur kawat yaitu *Vcc*, *Ground*, *Serial Clock (SCK)*, *Master-Output-Slave-Input (MOSI)*, *Master-Input-Slave-Output (MISO)*, *Slave-Select (SS)* dan beberapa kasus menggunakan pin *Reset (R)*. Namun pada dasarnya SPI hanya terdiri dari 3 pin, yaitu SCK, MOSI, dan MISO.



Gambar 3. Konsep Komunikasi SPI
(Sumber: Suhaeb, dkk., 2017:136)

Contoh komponen yang menggunakan komunikasi SPI sebagai antarmukanya adalah *RFID Reader* yaitu modul MFRC522, modul MicroSD, *downloader* mikrokontroler, dan sebagainya. Kecepatan transfer data pada SPI dapat mencapai 10Mbps.

C. Alat dan Bahan

1. Sistem Pengaturan Parkir berbasis RFID (Modul MicroSD dan Arduino Mega)
2. Kabel USB
3. Komputer dengan Software Arduino IDE yang sudah terpasang.

D. Keselamatan Kerja

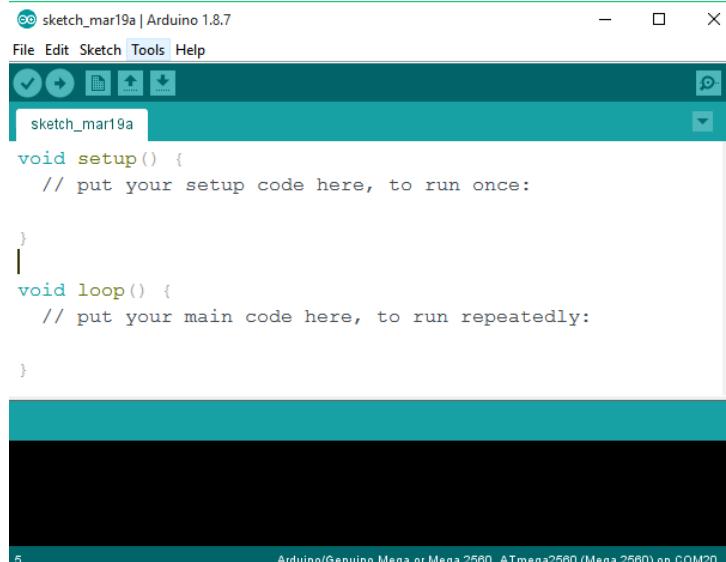
1. Pastikan saat mulai membuat rangkaian dalam kondisi *off* atau tanpa tegangan.
2. Gunakan alat dan bahan sesuai dengan fungsinya.
3. Jauhkan alat dan bahan yang tidak digunakan dari meja kerja.
4. Dalam merangkai pastikan sesuai dengan langkah kerja yang sudah ditentukan.
5. Tanyalah kepada pengampu apabila terjadi kesalahan dalam pengoperasian dan apabila terdapat penjelasan atau materi yang kurang dimengerti.

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL READER-WRITER MICROSD DENGAN ARDUINO MEGA	4x45 menit	
No. 02	Revisi: 01	Tgl: April 2019	Hal: 4 dari 11

E. Langkah Kerja

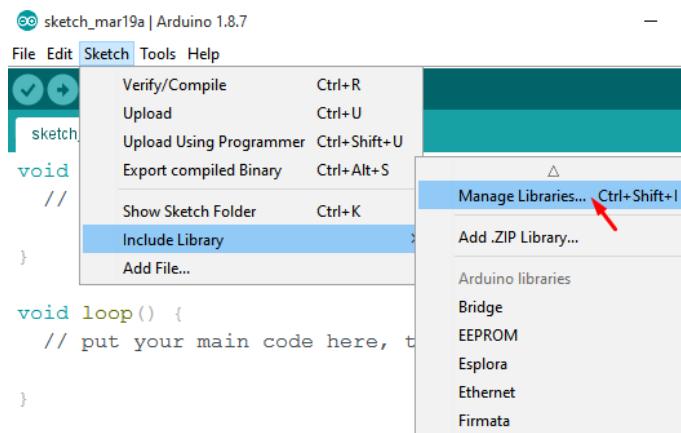
1. Memasang *library* SD pada Arduino IDE

- Buka *software* Arduino IDE, maka akan muncul tampilan utama.



Gambar 4. Tampilan awal Arduino IDE

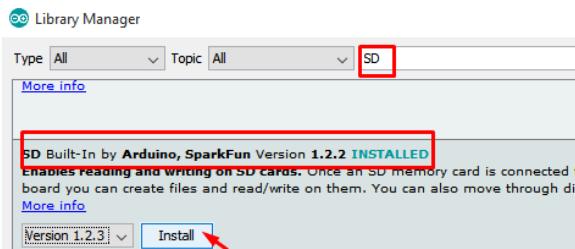
- Koneksikan Komputer ke internet kemudian pilih "Sketch" – pilih "Include Library" – pilih "Manage Libraries".



Gambar 5. Membuka jendela "Manage Libraries"

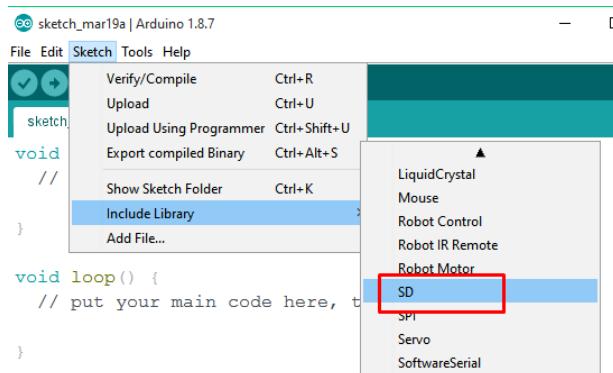
Setelah itu akan muncul jendela "Library Manager" – kemudian ketikan "SD" dan tekan "Enter" – cari yang dipublish oleh "Sparkfun" – Pilih versi terbaru – Klik "Install" – kemudian tunggu proses hingga selesai.

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL <i>READER-WRITER MICROSD</i> DENGAN ARDUINO MEGA	4x45 menit	
No. 02	Revisi: 01	Tgl: April 2019	Hal: 5 dari 11



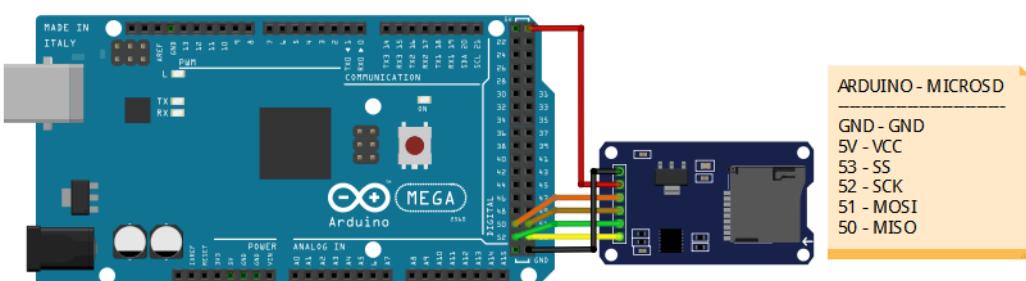
Gambar 6. Tampilan "Library Manager"

- c. Untuk menguji apakah *library* SD sudah masuk atau belum silakan klik "Sketch" – kemudian geser kursor ke "Include Library" – pastikan ada *library* SD di situ.



Gambar 7. Cek *library* SD

2. Mengidentifikasi rangkaian Arduino Mega dengan Modul MicroSD
Modul RTC dan Arduino Mega sudah tersambung dan terangkai dalam Sistem Pengaturan Parkir dengan koneksi seperti Gambar 8.



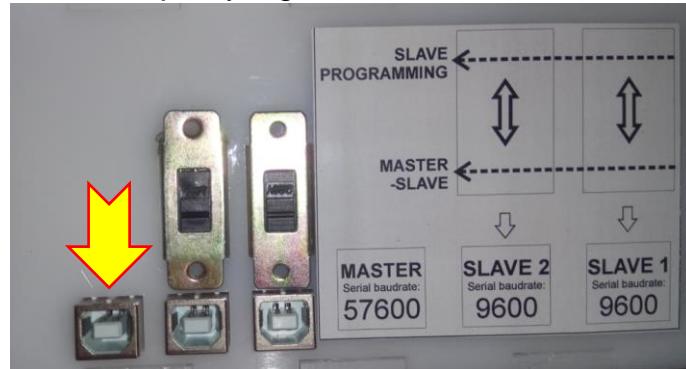
Gambar 8. Skematik pengabelan Modul MicroSD dengan Arduino Mega pada Sistem Pengaturan Parkir



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4	MENAKSES MODUL READER-WRITER MICROSD DENGAN ARDUINO MEGA	4x45 menit
No. 02	Revisi: 01	Tgl: April 2019

3. Menghubungkan Arduino Mega (master) pada Sistem Pengaturan Parkir dengan komputer melalui *port* yang sudah tersedia.



Gambar 9. Konektor USB ke master

4. Buka penutup *Main Controller* Sistem Pengaturan Parkir kemudian ambil SDCARD dan buka menggunakan komputer. Buat folder pada SDCARD bernama “DATABASE” dan “LOG”, dalam folder “DATABASE” buatlah *file* yang bernama “DATABASE.txt” yang berisi:

```
/*
 *INI ADALAH DATABASE SISTEM PARKIR
 *FORMAT DATABASE = NAMAPEMILIK-RFID
 *CONTOH = ARI_SANTOSO XX494FXX
 */
MASTER 8C59EF49
MASTER 6BAC95B9
ARI 49A63B2D
BUDI 24D2E920
CECEP 49FB852D
DENI 395A612D
ENI 497FE52D
FANI 1408B120
GHANI 399B362D
HANI 493A1F2D
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENGAKSES MODUL
READER-WRITER MICROSD
DENGAN ARDUINO MEGA

4x45 menit

No. 02

Revisi: 01

Tgl: April 2019

Hal: 7 dari 11

ISNAN 39F6172D

JANI 49DBA72D

KIKI 39245E2D

LALA 39D6BC2D

MERI 49B80F2D

NENI 4977A82D

5. Memprogram dengan Arduino Ide.

Buka Arduino IDE kemudian isikan program di bawah ini:

```
#include <SPI.h>
#include <SD.h>
File myFile;
String str1,str2,str3;
void writeToLOG(String a="")
{
    myFile = SD.open("LOG/log.txt", FILE_WRITE);
    // jika file bisa dibuka, kemudian tulis kedalam file.
    if (myFile)
    {
        myFile.println(a);
        // tutup file
        myFile.close();
        Serial.print(a);
        Serial.println(" has been written to log.");
    } else
    {
        // jika file tidak bisa dibuka, tampilkan pesan eror
        Serial.println("error opening log.txt");
    }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENGAKSES MODUL
READER-WRITER MICROSD
DENGAN ARDUINO MEGA

4x45 menit

No. 02

Revisi: 01

Tgl: April 2019

Hal: 8 dari 11

```
String readDatabase(String a="")  
{  
    String result="";  
    String database="";  
    unsigned int found=0, nameStart=0;  
    myFile = SD.open("DATABASE/DATABASE.txt");  
    if (myFile)  
    {  
        while (myFile.available())  
        {  
            database=myFile.readString();  
            found=database.indexOf(a);  
            if(found!=-1)  
            {  
                nameStart=database.lastIndexOf("\n", found);  
                result=database.substring(nameStart+1,found-1);  
            }  
            else  
            {  
                result="";  
            }  
            found=0;  
        }  
        myFile.close();  
    } else {  
        // jika file tidak bisa dibuka, tampilkan pesan eror  
        result="DATABASE ERROR";  
    }  
    return result;  
}  
void setup() {  
    str1="";
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENGAKSES MODUL
READER-WRITER MICROSD
DENGAN ARDUINO MEGA

4x45 menit

No. 02

Revisi: 01

Tgl: April 2019

Hal: 9 dari 11

```
str2="";
str3="";
// Menghidupkan komunikasi serial pada baudrate 9600
Serial.begin(9600);
while (!Serial) {
    ;
}
Serial.print("Initializing SD card...");
if (!SD.begin(53)) {
    Serial.println("SD CARD failed!");
    while (1);
}
Serial.println("initialization done.");
// Buka file. Satu file tidak bisa dibuka satu waktu.
// Jadi Anda harus menutupnya terlebih dahulu
// jika akan memproses kembali.
myFile = SD.open("test.txt", FILE_WRITE);
// jika file bisa dibuka, tulis pada file:
if (myFile) {
    Serial.print("Writing to test.txt...");
    myFile.println("testing 1, 2, 3.");
    // tutup file.
    myFile.close();
    Serial.println("done.");
} else
{
    // jika file tidak bisa dibuka, tampilkan pesan error
    Serial.println("error opening test.txt");
}
// membuka kembali file untuk pembacaan
myFile = SD.open("test.txt");
if (myFile) {
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENGAKSES MODUL
READER-WRITER MICROSD
DENGAN ARDUINO MEGA

4x45 menit

No. 02

Revisi: 01

Tgl: April 2019

Hal: 10 dari 11

```
Serial.println("test.txt:");
while (myFile.available()) {
    Serial.write(myFile.read());
}
// close the file:
myFile.close();
} else {
// jika file tidak bisa dibuka, tampilkan pesan eror
    Serial.println("error opening test.txt");
}
SD.mkdir("LOG");
}

void loop() {
if(Serial.available())
{
    str1=Serial.readStringUntil(' ');
    str2=Serial.readStringUntil(' ');
    str3=Serial.readStringUntil(' ');
}
if(str1 != "" && str2 != "" && str3 != "")
{
    str2=readDatabase(str2);
    Serial.println(str1);
    Serial.println(str2);
    Serial.println(str3);
    writeToLOG(str1);
    str1="";
    str2="";
    str3="";
}
}
```

6. *Upload program ke perangkat.*

Dibuat oleh:
Isnain Nabawi

Dilarang memperbanyak sebagian atau seluruh isi dokumen
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh:



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4	MENGAKSES MODUL <i>READER-WRITER MICROSD DENGAN ARDUINO MEGA</i>	
No. 02	Revisi: 01	Tgl: April 2019

7. Jalankan serial monitor melalui Arduino IDE.
8. Kirim perintah ke Arduino melalui serial monitor kemudian lihat hasilnya dan masukkan pada tabel di bawah ini:

No	Kirim perintah	<i>Output serial monitor</i>	Isi file LOG/log.txt
1	(tidak ada perintah)		
2	ABCDEF		
3	MASTER		
4	ARI		
5	6BAC95B9		
6	49A63B2D		
7	7711DB77		

F. Bahan Diskusi

Rangkai Arduino Mega dan Modul MicroSD. Buatlah file di dalam folder “DATABASE” yang bernama “(AWAL NAMA ANDA).txt” yang berisi teks dari NIM Anda dan NIM 2 teman Anda (Misal Anda bernama Isnabawani dengan NIM 15518241002 berarti Anda membuat file di dalam folder “DATABASE” yang bernama “ISNAN.txt” yang berisi teks “15518241002 15518241001 15518241003”). Buatlah program cek isi file dengan Arduino dengan keluaran: ketika Anda mengetikkan NIM Anda atau NIM 2 teman Anda lainnya pada serial monitor, maka muncul kalimat “TRUE” dalam serial monitor, jika tidak maka akan muncul tulisan “FALSE” pada serial monitor!

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit
	No. 04	Revisi: 01	Tgl: April 2019 Hal: 1 dari 30

A. Tujuan

1. Mengetahui rancangan sistem elektronik pada Sistem Pengaturan Parkir Berbasis RFID
2. Mengetahui cara memprogram Sistem Pengaturan Parkir Berbasis RFID
3. Mengetahui cara pengoperasian Sistem Pengaturan Parkir Berbasis RFID

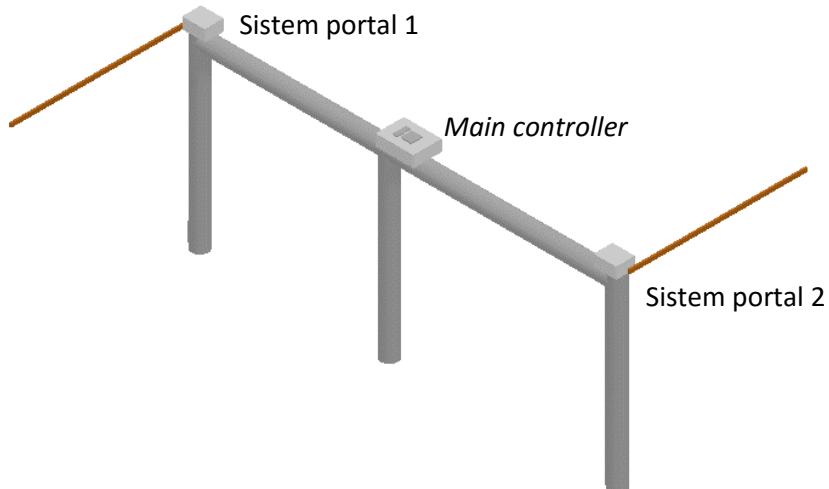
B. Dasar Teori

1. Sistem Pengaturan Parkir Berbasis RFID

Sistem Pengaturan Parkir berbasis RFID ini adalah suatu sistem pengelolaan parkir yang digunakan pada fasilitas parkir khusus. Sistem Pengaturan Parkir ini menggunakan kartu identitas khusus yang diberikan kepada setiap pengguna parkir. Kartu identitas ini menggunakan teknologi RFID. Sistem ini memiliki sistem pengamanan dan pembatasan gerak pengguna. Sistem pengamanan berupa pemberian hak parkir hanya pengguna yang memiliki kartu parkir dan terdaftar dalam sistem. Pembatasan gerak tersebut berupa waktu parkir dan kuota parkir.

Sistem parkir ini menggunakan mikrokontroller sebagai pemrosesan utamanya. Sistem parkir ini memiliki tiga bagian. Bagian pertama adalah *main controller*, bagian kedua dan ketiga adalah sistem portalnya. *Main controller* berisi kontroler utama berupa sistem *master-slave* Arduino Mega dan Arduino Nano, pembaca RFID, layar informasi, serta *keypad*. Sistem portal berisi *motor dc*, *driver motor dc*, serta sensor ultrasonik. Motor dc beserta *driver*-nya digunakan sebagai sistem penggerak palang sedangkan sensor ultrasonik digunakan sebagai sensor lewat kendaraan. Antara *main controller* dengan sistem portal dihubungkan melalui koneksi berkabel (*wired connection*).

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit	
No. 04	Revisi: 01	Tgl: April 2019	Hal: 2 dari 30



Gambar 1. Desain Posisi *Main Controller* dengan Sistem Portal

Boks *main controller* berukuran panjang 20cm, lebar 15cm, dan tinggi 8 cm. Boks sistem portal memiliki ukuran panjang 12cm, lebar 10cm, tinggi 9cm, dan panjang portal 100cm. Masing-masing boks berbahan akrilik yang memiliki tebal 5mm dan palang berbahan dasar aluminium. Spesifikasi hardware dari sistem pengaturan parkir berbasis RFID adalah sebagai berikut:

- a. Arduino Mega 2560 sebagai *master*.
- b. Arduino Nano R3 sebagai *slave*.
- c. Pembaca RFID dan kartu RFID sebagai simulasi pengguna parkir dengan frekuensi 13,56 MHz.
- d. *Real-time-clock* dengan IC DS3231 sebagai simulasi pembatasan waktu parkir.
- e. *Limit switch*, sensor *ultrasonic* HC-SR04 dan motor dc dengan torsi 10Kgcm sebagai miniatur palang pada pintu masuk parkir.
- f. Modul microSD sebesar 16 Gb sebagai penyimpan basis data.
- g. *Liquid Crystal Display (LCD)* 16x2 sebagai penyampai informasi alfa-numerik.
- h. *Keypad 4x4* sebagai opsi masukan.



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

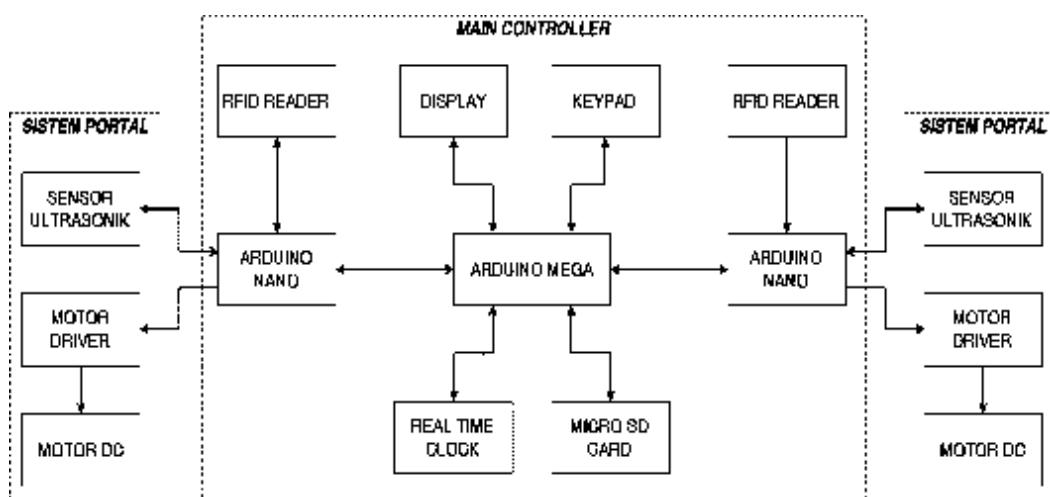
No. 04

Revisi: 01

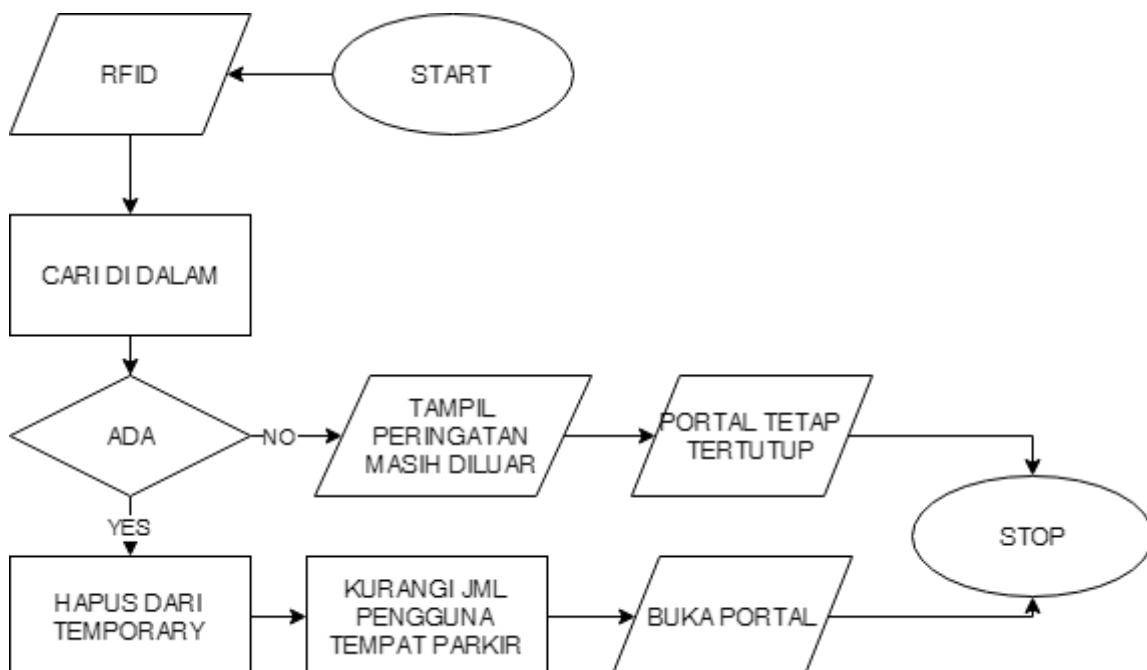
Tgl: April 2019

Hal: 3 dari 30

Secara keseluruhan penggunaan komponen dan hubungan antar komponen dapat dilihat pada Gambar 2, kemudian untuk jalannya program dapat diidentifikasi melalui flowchart yang terdapat pada Gambar 3 dan Gambar 4.



Gambar 2. Bagan Sistem



Gambar 3. Flowchart saat pengguna keluar



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

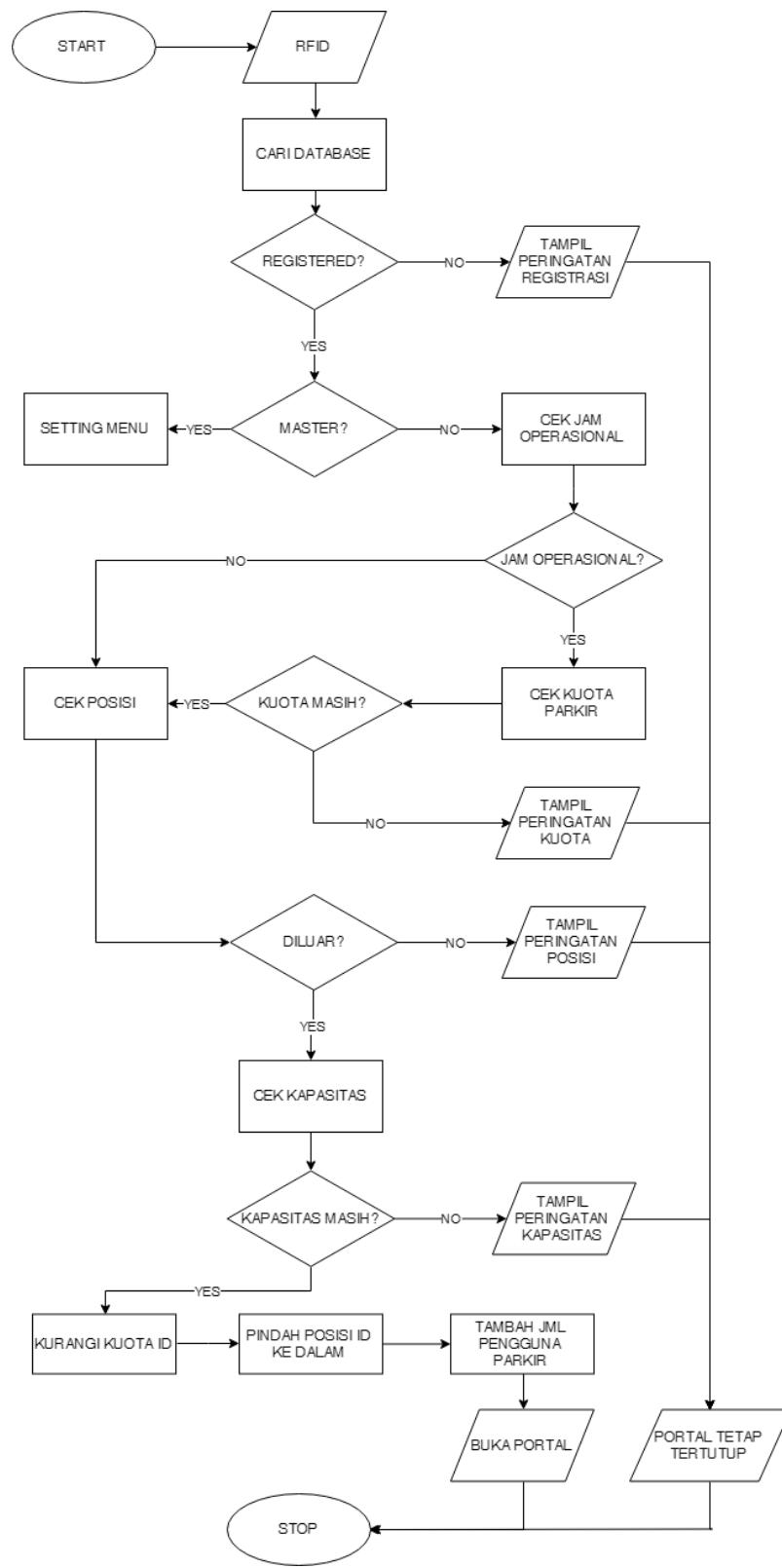
4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 4 dari 30



Gambar 4. Flowchart saat pengguna masuk.

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit
	No. 04	Revisi: 01	Tgl: April 2019 Hal: 5 dari 30

Sistem elektronik didesain berdasarkan bagan yang sudah dibuat. Bagan sistem dapat dilihat pada Gambar 2. Untuk mempermudah maka bagan akan dibagi menjadi 3 bagian pembahasan yaitu Master, Slave, dan Portal. Selain itu ada bagian power unit serta desain *printed circuit board* (PCB).

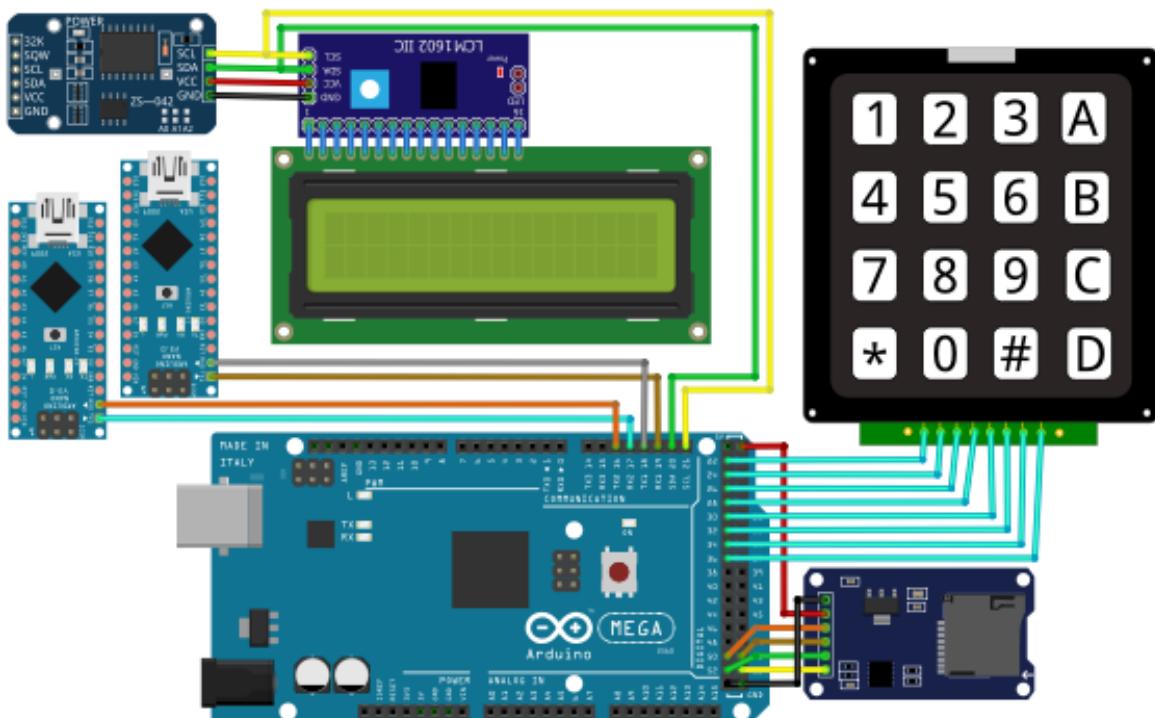
2. Master

Sistem kontrol utama pada master menggunakan mikrokontroler ATmega 2560 yang sudah menjadi modul berupa Arduino Mega. Arduino Mega bertanggung jawab untuk memproses data dari *Real Time Clock* (RTC), *Micro SD Card*, *Keypad 4x4* serta memberikan informasi melalui *Liquid Crystal Display* (LCD). Modul RTC berbasis pada IC DS3231 dan sudah memiliki baterai sendiri sehingga jika power unit dari sistem mati/tidak tersedia, waktu pada modul akan tetap akurat. LCD berperan sebagai *display* dengan ukuran 16 karakter x 2 baris, memiliki *background* hijau, *backlight* dan sudah dilengkapi dengan modul LCD *driver Inter-Integrated Circuit* (I2C).

Arduino Mega 2560 berkomunikasi dengan RTC dan LCD menggunakan komunikasi I2C. RTC menggunakan alamat *default* yaitu 0x27 sementara LCD menggunakan alamat 0x26. Pin SDA dan SCL pada RTC dan LCD masing-masing terhubung pada pin 20 dan 21 pada Arduino Mega 2560. Keypad 4x4 terhubung pada Arduino Mega 2560 pin 30, 32, 34, dan 36 untuk kolom dan pin 22, 24, 26, dan 28 untuk baris. Komunikasi Serial *Universal Asynchronous Receiver and Transmitter* (UART) nantinya akan digunakan untuk komunikasi antar Arduino Nano yang bertindak sebagai kontrol utama *slave* dengan Arduino Mega 2560 yang bertindak sebagai *master*. Slave 1 yang bertindak sebagai pelayan masuk menggunakan jalur

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
	SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
	No. 04	Revisi: 01	Tgl: April 2019	Hal: 6 dari 30

komunikasi Serial1 dengan pin 18 sebagai Tx1 dan 19 sebagai Rx1. Slave 2 yang bertindak sebagai pelayan keluar menggunakan jalur komunikasi Serial2 dengan pin 16 sebagai Tx2 dan 17 sebagai Rx2. Sesuai dengan pengabelan komunikasi serial UART, pin Tx pada slave tersambung pada Rx master dan pin Rx pada slave tersambung pada Tx master. Modul MicroSD terhubung menggunakan jalur komunikasi *Serial Peripheral Interface* (SPI). Pada Arduino Mega 2560 jalur komunikasi SPI berada pada pin 53 sebagai SS, pin 52 sebagai SCK, pin 51 sebagai MOSI, dan pin 50 sebagai MISO.



Gambar 5. Skema Elektronik pada bagian Master

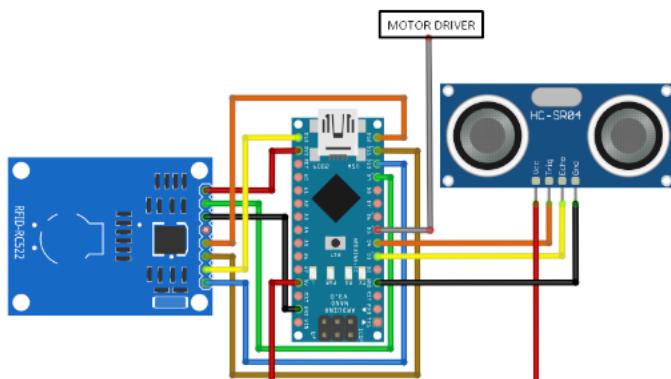
3 Slave

Slave berbasis pada IC ATmega 328p yang sudah menjadi modul Arduino Nano. Arduino Nano bertanggung jawab untuk melakukan pembacaan RFID, membaca jarak dengan sensor ultrasonik, serta membuka dan menutup portal. Arduino Nano

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
No. 04	Revisi: 01	Tgl: April 2019	Hal: 7 dari 30

menggunakan pembacaan RFID menggunakan modul RFID-RC522 melalui jalur komunikasi SPI. Jalur komunikasi SPI pada Arduino Nano berada pada pin D12 sebagai MISO, pin D11 sebagai MOSI, pin D13 sebagai SCK, dan pin D10 sebagai SDA.

Arduino Nano melakukan pembacaan jarak menggunakan modul sensor HC-SR04 melalui 2 pin yaitu *echo* dan *trig*. Pin *echo* terhubung pada pin D3 dan *trig* terhubung pada pin D4. Arduino Nano melakukan proses buka tutup portal cukup menggunakan 1 jalur yaitu D5 yang terhubung pada *motor driver*. Sinyal *high* untuk membuka portal dan sinyal *low* untuk menutup portal.



Gambar 6. Skema Elektronik pada bagian *Slave*

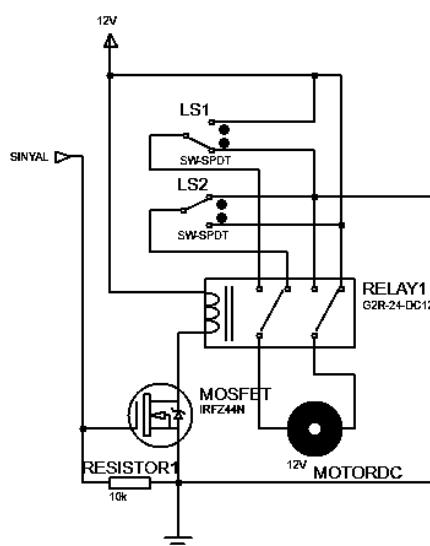
4. Portal

Portal terdiri dari *motor driver* dan motor dc. Motor dc yang dipakai adalah motor dc dengan *gearbox*. Spesifikasi motor dc yang digunakan memiliki tegangan 12V DC dengan kecepatan 12 RPM dan torsi 10Kg/cm. Alasan penggunaan motor tersebut adalah karena motor dc nantinya tidak memerlukan satu putaran penuh sehingga cukup membutuhkan RPM yang rendah namun memiliki torsi yang besar. Sistem juga

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
	SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
	No. 04	Revisi: 01	Tgl: April 2019	Hal: 8 dari 30

diharuskan memiliki pembatasan menggunakan *limit switch* untuk posisi portal terbuka dan tertutup.

Motor *driver* yang digunakan didesain khusus agar dapat menyesuaikan dengan kebutuhan. Kebutuhan yang dimaksud adalah motor *driver* harus dapat membaca 1 jalur sinyal *high* atau *low* saja, dapat membalik-balikkan putaran motor, bekerja dengan *limit switch*, dapat melakukan penggereman, serta tidak memerlukan pengaturan kecepatan. Agar dapat membaca 1 jalur saja maka motor *driver* menggunakan relai yang telah dikuatkan oleh transistor Mosfet IRFz44. Relai yang digunakan adalah berjenis *double pole double throw* (DPDT). *Limit switch* yang digunakan adalah berjenis *single pole double throw* (SPDT). Relai digunakan untuk membalik-balikkan putaran motor berdasarkan sinyal dari *slave*. *Limit switch* digunakan untuk melakukan pemutusan tegangan ke motor serta menghubung-singkatkan terminal motor agar motor dapat melakukan penggereman dan berhenti seketika.



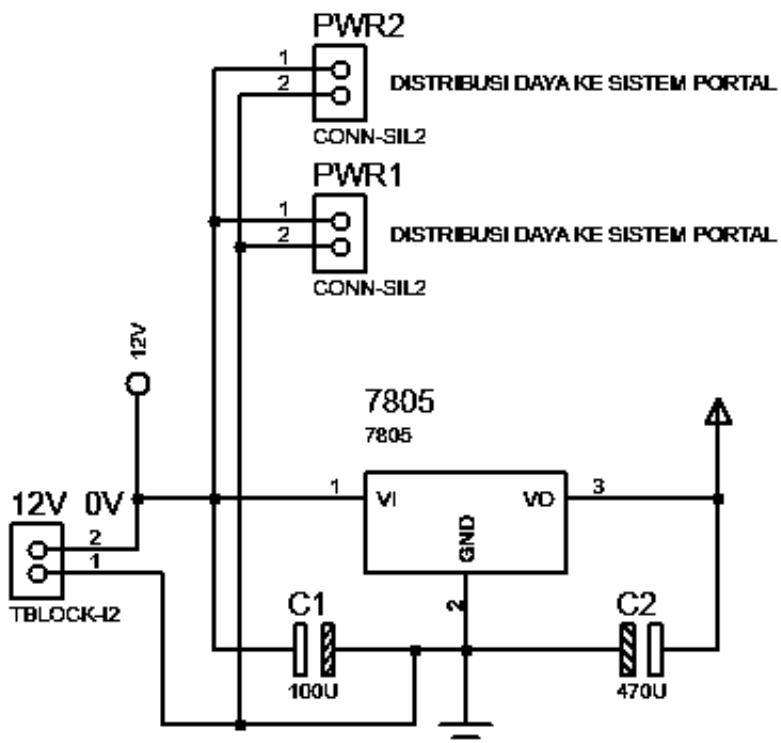
Gambar 7. Skema elektronik motor driver

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
No. 04	Revisi: 01	Tgl: April 2019	Hal: 9 dari 30

5. Power Unit

Sistem pengaturan parkir berbasis RFID ini disuplai oleh adaptor DC 12V.

Adaptor tersebut kemudian dibagi tegangannya menjadi 12V, 5V, dan 3,3V. Adaptor yang digunakan memiliki kemampuan menyuplai arus sebesar 1000mA. Tegangan 12V menyuplai *motor driver* dan motornya itu sendiri. Tegangan 5V dihasilkan dari IC regulator 7805 yang digunakan untuk menyuplai kontroler dan komponen lainnya kecuali modul RFID-RC522 yang menggunakan tegangan 3,3V yang dihasilkan dari IC regulator tegangan AMS 1117 pada modul Arduino Nano. IC regulator 7805 memiliki kemampuan menyuplai arus hingga 1000mA. IC regulator tegangan AMS 1117 memiliki kemampuan menyuplai arus hingga 1000mA.



Gambar 8. Skema elektronik sistem distribusi daya pada *main board*

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
No. 04	Revisi: 01	Tgl: April 2019	Hal: 10 dari 30

C. Alat dan Bahan

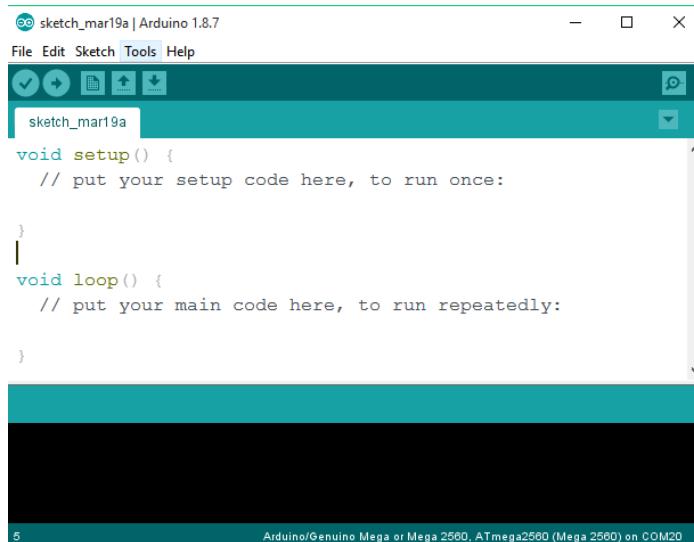
1. Sistem Pengaturan Parkir Berbasis RFID (*Main Controller* dan kedua sistem portal)
2. Kabel USB
3. Komputer dengan *Software Arduino IDE* yang sudah terpasang.
4. *Stopwatch*

D. Keselamatan Kerja

1. Pastikan saat memulai membuat rangkaian dalam kondisi *off* atau tanpa tegangan.
2. Gunakan alat dan bahan sesuai dengan fungsinya.
3. Jauhkan alat dan bahan yang tidak digunakan dari meja kerja.
4. Dalam merangkai pastikan sesuai dengan langkah kerja yang sudah ditentukan.
5. Tanyalah kepada pengampu apabila terjadi kesalahan dalam pengoperasian dan apabila terdapat penjelasan atau materi yang kurang dimengerti.

E. Langkah Kerja

1. Menambahkan beberapa *library* yang dibutuhkan pada Arduino IDE
 - a. Buka *software Arduino IDE*, maka akan muncul tampilan utama.



```
sketch_mar19a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_mar19a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

5
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM20
```

Gambar 9. Tampilan utama Arduino IDE

2. Memprogram Master/Arduino Mega dengan Arduino Ide.

Buka Arduino IDE kemudian isikan program di bawah ini:

Dibuat oleh: Isnabawi	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh:
--------------------------	---	-----------------



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 11 dari 30

```
#include <Wire.h> //library i2c
#include <LiquidCrystal_I2C.h> //library LCD
#include <DS3231.h> //library RTC
#include <SPI.h> //library komunikasi SPI
#include <SD.h> //library modul SDCARD
#include <Keypad.h> //library keypad

//inisialisasi keypad
const byte numRows= 4;
const byte numCols= 4;
char keymap[numRows][numCols]=
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'} };
byte rowPins[numRows] = {30,32,34,36}; //pin untuk baris
byte colPins[numCols] = {22,24,26,28}; //pin untuk kolom

Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);
//akhir inisialisasi keypad

//inisialisasi LCD dengan alamat 0x26, 16 karakter, dan 2 baris
LiquidCrystal_I2C lcd(0x26, 16, 2);

//inisialisasi file pada SDCard
File myFile;

//inisialisasi RTC
DS3231 Clock;
bool Century=false;
bool h12;
bool PM;
byte Year;
byte Month;
byte Date;
byte DoW;
byte Hour;
byte Minute;
byte Second;
//akhir inisialisasi RTC

String str1="",str2="",scnResult="";
byte usedQuota=0;
bool RFIDReady=0,RFIDOutReady=0;
bool masterSetting=0;

//inisialisasi syarat-syarat parkir
bool registeredID=0; //id sudah teregistrasi atau belum
bool workingHours=0; //jam kerja
bool parkingQuota=0; //kuota parkir
bool parkingSlots=0; //slot parkir
bool userPosition=0; //posisi ID, 1 ID tdk bisa digunakan oleh 2 kendaraan, harus
//keluar dahulu.
bool specialHours=0; //SPECIAL HOURS tidak pakai kuota di jam ini
//akhir inisialisasi syarat-syarat parkir

unsigned int userCounts=0,userMax=0,quotaMax=0,timeMin=0,timeMax=0;
```

Dibuat oleh:
Isnab Nabawi

Dilarang memperbanyak sebagian atau seluruh isi dokumen
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh:



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit
No. 04	Revisi: 01	Tgl: April 2019

```
void readSetting(unsigned int& userMax,unsigned int& quotaMax,unsigned int&
timeMin,unsigned int& timeMax)
{
    //Ini adalah fungsi untuk membaca pengaturan yang tersimpan
    String temp1="",userMaxS="",quotaMaxS="",timeMinS="",timeMaxS="";
    byte idxA,idxB,idxC,idxD;
    if (SD.exists("DATABASE/SETTING.txt"))
    {
        myFile = SD.open("DATABASE/SETTING.txt");
        while(myFile.available())
        {
            temp1 = myFile.readString();
            idxA = temp1.indexOf("A");
            idxB = temp1.indexOf("B");
            idxC = temp1.indexOf("C");
            idxD = temp1.indexOf("D");

            userMaxS = temp1.substring(0,idxA);
            quotaMaxS= temp1.substring(idxA+1,idxB);
            timeMinS = temp1.substring(idxB+1,idxC);
            timeMaxS = temp1.substring(idxC+1,idxD);

            userMax = userMaxS.toInt();
            quotaMax= quotaMaxS.toInt();
            timeMin = timeMinS.toInt();
            timeMax = timeMaxS.toInt();
        }
        myFile.close();
    }
    else
    {
        //tampilan error jika file setting tidak ditemukan/rusak
        lcd.setCursor(0,0);
        lcd.print("BASE SETTING ERR");
        while(1){}
    }
}

void writeSetting(unsigned int userX,unsigned int quotaX,
                  unsigned int timeN,unsigned int timeX)
{
    //ini adalah fungsi untuk mengubah pengaturan
    if (SD.exists("DATABASE/SETTING.txt"))
    {
        SD.remove("DATABASE/SETTING.txt");
        myFile = SD.open(String("DATABASE/SETTING.txt"), FILE_WRITE);
        myFile.print(String(userX));
        myFile.print("A");
        myFile.print(String(quotaX));
        myFile.print("B");
        myFile.print(String(timeN));
        myFile.print("C");
        myFile.print(String(timeX));
        myFile.print("D");
        Serial.println(timeX);
        //myFile.print(timeX);
        myFile.close();
    }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 13 dari 30

```
}

else
{
    lcd.setCursor(0,0);
    lcd.print("WRITE PARAM ERROR");
    while(1){}
}

}

String scanTemp(String a="")
{
    //ini adalah fungsi mencari kendaraan yang ada di dalam parkiran
    String result="";
    String temp="";
    String tgl="";
    bool tempOK=0;
    bool tempExist=0;
    unsigned int tanggal,tanggalLama=0;
    byte quota=0;
    unsigned int found=0,quoStart=0,foundDate=0,tglStart=0;

    tanggal=Clock.getDate();

    //special hours
    if(Clock.getHour(h12, PM)>=timeMin &&
       Clock.getHour(h12, PM)<timeMax)
        specialHours=0;
    else
        specialHours=1;
    //special hours

    if (SD.exists(String("TEMP/" + a + ".txt")))
    {
        myFile = SD.open(String("TEMP/" + a + ".txt"));
        while(myFile.available())
        {
            temp=myFile.readString();
            //Serial.println(temp);
            tgl=temp.substring(2);
            tanggalLama=tgl.toInt();
            //Serial.println(tgl);
            temp=temp.substring(0,1);

            quota=temp.toInt();
        }
        myFile.close();

        if(tanggalLama==tanggal && specialHours!=1)
        {
            quota++;if(quota>9)quota=9;
            //maksimum kuota terbaca =9
        }
        else
            quota=1;

        SD.remove(String("TEMP/" + a + ".txt"));
        myFile = SD.open(String("TEMP/" + a + ".txt"), FILE_WRITE);
        myFile.print(String(quota));
        myFile.print(" ");
        myFile.print(String(tanggal));
    }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 14 dari 30

```
        myFile.close();
    }
else
{
    myFile = SD.open(String("TEMP/" + a + ".txt"), FILE_WRITE);
    quota=1;
    myFile.print("1");
    myFile.print(" ");
    myFile.print(String(tanggal));
    myFile.close();
}
found=0;
return String(quota);
}

String scanDatabase(String a="")
{
    //ini adalah fungsi untuk mencari ID yang sudah terdaftar
    String result="";
    String database="";
    unsigned int found=0, nameStart=0;
    myFile = SD.open("DATABASE/DATABASE.txt");
    if (myFile)
    {
        while (myFile.available())
        {
            database=myFile.readString();
            found=database.indexOf(a);
            if(found!=-1)
            {
                nameStart=database.lastIndexOf("\n", found);
                result=database.substring(nameStart+1,found-1);
            }
            else
            {
                result="NOTFOUND";
            }
            found=0;
        }
        myFile.close();
    }
    else
    {
        result="DATABASEERROR";
    }
    return result;
}

unsigned int userComing(String a="")
{
    //ini adalah fungsi pencatatan ID yang masuk parkiran
    String temp;
    unsigned int user=0;

    //input user to userlist
    if(SD.exists("USERLIST"))
    {
        myFile = SD.open(String("USERLIST/" + a),FILE_WRITE);
        myFile.print("1");
        myFile.print(" ");
        myFile.print(String(tanggal));
        myFile.close();
    }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 15 dari 30

```
        myFile.close();
    }
else
{
    SD.mkdir("USERLIST");
    myFile = SD.open(String("USERLIST/" + a),FILE_WRITE);
    myFile.close();
}
return user;
}

unsigned int userOutgoing(String a="")
{
    //ini adalah fungsi pencatatan ID yang keluar parkiran
    String temp;
    unsigned int user=0;

    //delete userlist
    if(SD.exists(String("USERLIST/" + a)))
    {
        SD.remove(String("USERLIST/" + a));
    }
    else
    {
        Serial.println("HITUNGAN OUTGOING ERROR!");
        lcd.setCursor(0,1);
        lcd.print("COUNTS ERROR!");
        while(1){};
    }
    return user;
}

bool getUserList(String a="")
{
    //ini adalah fungsi pencatatan ID yang masuk parkiran
    bool userSignedIn=0;
    if(SD.exists("USERLIST"))
    {
        myFile = SD.open(String("USERLIST/" + a));
        if(myFile) userSignedIn=1;
        else userSignedIn=0;
        myFile.close();
    }
    else
    {
        lcd.setCursor(0,1);
        lcd.print("USERLIST FLDR ERROR");
        while(1){}
    }
    return userSignedIn;
}

unsigned int getUserCounts()
{
//ini adalah fungsi untuk menghitung jumlah kendaraan yang ada di dalam parkiran
    File d = SD.open("USERLIST");
    int count_files = 0;
    bool counting=1;
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 16 dari 30

```
while(counting)
{
  File entry =  d.openNextFile();
  if( !entry )
  {
    counting=0;
  }
  else
  {
    count_files++;
  }
}
d.close();
return count_files;
}
void setup() {

  Serial.begin(57600); //baudrate untuk Serial Monitor
  Serial1.begin(9600); //baudrate untuk SLAVE1
  Serial2.begin(9600); //baudrate untuk SLAVE2
  Wire.begin();
  lcd.begin();

  //menghidupkan LCD
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Initialize...");
  delay(500);

  //cek SDCard
  if (!SD.begin(53))
  {
    lcd.setCursor(0,0);
    lcd.print("SD CARD ERROR!    ");
    while (1){};
  }
  else
  {
    //cek default file yang harus ada dalam SDCard
    myFile=SD.open("USERLIST");
    if(!myFile)
    {
      lcd.setCursor(0,0);
      lcd.print("USERLIST ERROR!  ");
      lcd.setCursor(0,1);
      lcd.print("CHECK SD CARD    ");
      while(1){};
    }
    myFile.close();

    myFile=SD.open("TEMP");
    if(!myFile)
    {
      lcd.setCursor(0,0);
      lcd.print("TEMPORARY ERROR! ");
      lcd.setCursor(0,1);
      lcd.print("CHECK SD CARD    ");
      while(1){};
    }
  }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 17 dari 30

```
}

myFile.close();

myFile=SD.open("DATABASE/SETTING.txt");
if(!myFile)
{
    lcd.setCursor(0,0);
    lcd.print("BASE SETTING      ");
    lcd.setCursor(0,1);
    lcd.print("PARAMETER ERROR!");
    while(1){};
}
myFile.close();

myFile=SD.open("DATABASE/DATABASE.txt");
if(myFile)
{
    lcd.setCursor(0,0);
    lcd.print("DATABASE OK      ");
    myFile.close();
}
else
{
    lcd.setCursor(0,0);
    lcd.print("DATABASE ERROR!      ");
    while(1){};
}

readSetting(userMax,quotaMax,timeMin,timeMax);
Serial.print("Maksimum pengguna : ");
Serial.println(userMax);
lcd.setCursor(0,0);
lcd.print("KAPASITAS MAKS: ");
lcd.setCursor(0,1);
lcd.print(userMax);
delay(1000);
lcd.clear();

Serial.print("Kuota parkir maksimum pada jam operasional : ");
Serial.println(quotaMax);
lcd.setCursor(0,0);
lcd.print("KUOTA MAKS:      ");
lcd.setCursor(0,1);
lcd.print(quotaMax);
delay(1000);
lcd.clear();
Serial.print("Jam Kerja : Pukul ");
Serial.print(timeMin);
Serial.print(" : ");
Serial.print(timeMax);
Serial.println(" WIB");

lcd.setCursor(0,0);
lcd.print("JAM OPERASIONAL: ");
lcd.setCursor(0,1);
lcd.print(timeMin);
lcd.print(".00 - ");
lcd.print(timeMax);
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 18 dari 30

```
lcd.print(".00");
delay(1000);
//lcd.clear();
}
lcd.clear();
}

void loop() {

//PINTU MASUK
if(Serial1.available())
{
    str1=Serial1.readStringUntil('\n');
    str1=str1.substring(0,str1.length()-1);
    Serial.println(str1);
    if(str1=="SLAVEREADY") { RFIDReady=0; lcd.clear(); }
    else RFIDReady=1;
}

if(RFIDReady)
{
    scnResult=scanDatabase(str1);
    Serial.println(scnResult);
    if(scnResult != "NOTFOUND" && scnResult != "MASTER")
    {
        registeredID=1;
    }
    else if(scnResult == "MASTER")
    {
        //lcd.setCursor(0,1);
        //lcd.print(scnResult);
        masterSetting=1;
        masterEvent();
        registeredID=0;
    }
    else
    {
        registeredID=0;
    }
}

if(registeredID && workingHours)
{
    userCounts=getUserCounts();
    if(userCounts>=userMax) parkingSlots=0;
    else parkingSlots=1;

    if(getUserList(str1)) userPosition=0;
    else userPosition=1;
}

if(registeredID && workingHours && parkingSlots && userPosition)
{
    usedQuota=scanTemp(str1).toInt();
    if(usedQuota>quotaMax) parkingQuota=0;
    else parkingQuota=1;
    Serial.print("READED :");
    Serial.println(String(usedQuota));
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit
No. 04	Revisi: 01	Tgl: April 2019

```
lcd.setCursor(0,1);
if(registeredID && workingHours && parkingQuota && parkingSlots && userPosition)
{
    registeredID=0;
    userPosition=0;
    Serial.println("TRUE");
    Serial1.println("TRUE");
    lcd.print("SELAMAT DATANG  ");
    userCounts=userComing(str1);
    //delay(2000);
    //lcd.clear();
}
else
{
    if(registeredID==0) lcd.print("ID TDK TERDAFTAR");
    if(userPosition==0) lcd.print("ANDA MSH DIDALAM");
    if(parkingSlots==0) lcd.print("PARKIR PENUH      ");
    if(parkingQuota==0) lcd.print("KUOTA ANDA HABIS");
    if(workingHours==0) lcd.print("PARKIR 06-19 WIB");

    Serial.println("FALSE");
    Serial1.println("FALSE");
    //delay(2000);
    //lcd.clear();
}
str1="";
RFIDReady=0;
}

//PINTU KELUAR
if(Serial2.available())
{
    str2=Serial2.readStringUntil('\n');
    str2=str2.substring(0,str2.length()-1);
    Serial.println(String("SLAVE2 : " + str2));
    if(str2=="SLAVEREADY") { RFIDOutReady=0; lcd.clear(); }
    else RFIDOutReady=1;
}
if(RFIDOutReady)
{
    if(getUserList(str2))
    {
        userOutgoing(str2); //mengurangi
        Serial.println("SLAVE2: TRUE");
        Serial2.println("TRUE");
        lcd.setCursor(0,1);
        lcd.print("  SAMPAI JUMPA  ");
    }
    else
    {
        Serial.println("SLAVE2: FALSE");
        Serial2.println("FALSE");
        lcd.setCursor(0,1);
        lcd.print("ANDA MSH DILUAR ");
    }
    RFIDOutReady=0;
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 20 dari 30

```
showClock(0);
}
void showClock(bool a)
{
    lcd.setCursor(0,a);
    lcd.print(Clock.getHour(h12, PM));
    if((Clock.getSecond() % 2) == 0) lcd.print(":");
    else lcd.print(" ");
    lcd.print(Clock.getMinute());
    lcd.print(" ");
    lcd.print(Clock.getDate());
    lcd.print("/");
    lcd.print(Clock.getMonth(Century));
    lcd.print("/");
    lcd.print(Clock.getYear());
    lcd.print("    ");
}
void masterEvent()
{
    //fungsi ID master, menampilkan menu dan settingan khusus.
    while(masterSetting)
    {
        String inString="",stringTemp;
        byte count=0;
        bool singleAction=1;
        byte optionSelect=0;
        byte menuLevel=1;
        char keypressed;
        char timeSet[11];
        byte Temp1, Temp2;
        char temp[5];
        while(menuLevel==1)
        {
            lcd.setCursor(0,0);
            lcd.print("*MASTER SETTING*");
            keypressed = myKeypad.getKey();
            if (keypressed != NO_KEY)
            {
                if(keypressed=='*')
                {
                    masterSetting=0; menuLevel=0; lcd.clear();
                }
                else
                {
                    lcd.setCursor(0,1);
                    if      (keypressed=='1') { lcd.print("1.BUKA PORTAL    ");
                                              optionSelect=1; }
                    else if(keypressed=='2') { lcd.print("2.SET WAKTU      ");
                                              optionSelect=2; }
                    else if(keypressed=='3') { lcd.print("3.SET KUOTA MAKSA");
                                              optionSelect=3; }
                    else if(keypressed=='4') { lcd.print("4.SET KAPASITAS ");
                                              optionSelect=4; }
                    else if(keypressed=='5') { lcd.print("5.SET JAM KERJA  ");
                                              optionSelect=5; }
                    else if(keypressed=='#') { menuLevel=2; lcd.clear(); count=0; }
                    else
                        { lcd.print(" *MENU TDK ADA* ");
                          optionSelect=0; }
                }
            }
        }
    }
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 21 dari 30

```
        }
    }
}
while(menuLevel==2 && optionSelect== 1)
{
    if(singleAction)
    {
        Serial1.println("MASTER");
        Serial2.println("MASTER");
        singleAction=0;
    }
    lcd.setCursor(0,0);
    lcd.print("=PORTAL TERBUKA=");
    lcd.setCursor(0,1);
    lcd.print("EXIT UTK MENUTUP");
    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY)
    {
        if(keypressed=='*')
        {
            Serial1.println("FALSE");
            Serial2.println("FALSE");
            menuLevel=1;
            optionSelect=0;
            lcd.clear();
        }
    }
}
while(menuLevel==2 && optionSelect== 2)
{
    lcd.setCursor(0,0);
    lcd.print("YYMMDDHHMM #ok");
    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY)
    {
        if(keypressed=='*')
        {
            menuLevel=1;optionSelect=0;lcd.clear();
        }
        else if (count<10 && keypressed!='A' &&
keypressed!='B' && keypressed!='C' &&
keypressed!='D' && keypressed != '#')
        {
            inString=String(inString + keypressed);
            lcd.setCursor(0,1);
            lcd.print(inString);
            count++;
        }
        else if(keypressed == 'A')
        {
            count=0;
            inString="";
            lcd.clear();
        }
        else if(keypressed == '#' && count==10)
        {
            count=0;
            lcd.clear();
        }
    }
}
```



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4	RANCANG-BANCON SISTEM PENGATURAN PARKIR BERBASIS RFID		4x45 menit
	No. 04	Revisi: 01	Tgl: April 2019

```

Clock.setClockMode(false); // set to 24h
//setClockMode(true); // set to 12h
inString.toCharArray(timeSet, 11);

Temp1 = (byte)timeSet[0] -48;
Temp2 = (byte)timeSet[1] -48;
Year = Temp1*10 + Temp2;

Temp1 = (byte)timeSet[2] -48;
Temp2 = (byte)timeSet[3] -48;
Month = Temp1*10 + Temp2;

Temp1 = (byte)timeSet[4] -48;
Temp2 = (byte)timeSet[5] -48;
Date = Temp1*10 + Temp2;

Temp1 = (byte)timeSet[6] -48;
Temp2 = (byte)timeSet[7] -48;
Hour = Temp1*10 + Temp2;

Temp1 = (byte)timeSet[8] -48;
Temp2 = (byte)timeSet[9] -48;
Minute = Temp1*10 + Temp2;

Clock.setYear(Year);
Clock.setMonth(Month);
Clock.setDate(Date);
Clock.setHour(Hour);
Clock.setMinute(Minute);
Clock.setSecond(00);

lcd.setCursor(0,0);
lcd.print("OK! *Exit");
showClock(1);
menuLevel=1;optionSelect=0;
}

}

}

while(menuLevel==2 && optionSelect== 3) //SET KUOTA MAKSIMAL
{
//String inString="";
//char keypressed;
lcd.setCursor(0,0);
lcd.print("KUOTA MAKS @ID:");
keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
    if(keypressed=='*')
    {
        menuLevel=1;optionSelect=0;lcd.clear();
    }
    else if (keypressed!= 'A' && keypressed!= 'B' &&
              keypressed!= 'C' && keypressed!= 'D' && keypressed != '#')
    {
        inString=String(inString + keypressed);
        lcd.setCursor(0,1);
        lcd.print(inString);
    }
}
}

```



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 23 dari 30

```
else if (keypressed == '#')
{
    lcd.setCursor(0,1);
    if(inString != "")
    {
        quotaMax=inString.toInt();
        lcd.print("TERSIMPAN!      ");
    }
    else
    {
        lcd.print("TDK TERSIMPAN!  ");
    }
    writeSetting(userMax,quotaMax,timeMin,timeMax);
    menuLevel=1;optionSelect=0;
}
}
while(menuLevel==2 && optionSelect== 4) //SET KAPASITAS PARKIR
{
    //String inString="";
    //char keypressed;
    lcd.setCursor(0,0);
    lcd.print("KAPASITAS PARKIR:");
    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY)
    {
        if(keypressed=='*')
        {
            menuLevel=1;optionSelect=0;lcd.clear();
        }
        else if (keypressed!= 'A' && keypressed!= 'B' &&
                  keypressed!= 'C' && keypressed!= 'D' && keypressed != '#')
        {
            inString=String(inString + keypressed);
            lcd.setCursor(0,1);
            lcd.print(inString);
        }
        else if (keypressed == '#')
        {
            lcd.setCursor(0,1);
            if(inString != "")
            {
                userMax=inString.toInt();
                lcd.print("TERSIMPAN!      ");
            }
            else
            {
                lcd.print("TDK TERSIMPAN!  ");
            }
            writeSetting(userMax,quotaMax,timeMin,timeMax);
            menuLevel=1;optionSelect=0;
        }
    }
}
while(menuLevel==2 && optionSelect== 5) //SET JAM OPERASIONAL
{
    lcd.setCursor(0,0);
    lcd.print("HHHH CONTOH:0619");
}
```



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 24 dari 30

```
keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
    if(keypressed=='*')
    {
        menuLevel=1;optionSelect=0;lcd.clear();
    }
    else if (count<4 && keypressed!= 'A' && keypressed!= 'B' &&
              keypressed!= 'C' && keypressed!= 'D' && keypressed != '#')
    {
        count++;
        inString=String(inString + keypressed);
        lcd.setCursor(0,1);
        lcd.print(inString);
    }
    else if(keypressed == 'A')
    {
        count=0;
        inString="";
        lcd.clear();
    }
    else if (keypressed == '#' && count==4)
    {
        count=0;
        lcd.setCursor(0,1);
        if(inString != "")
        {
            inString.toCharArray(temp, 5);

            Temp1 = (byte)temp[0] -48;
            Temp2 = (byte)temp[1] -48;
            timeMin = Temp1*10 + Temp2;

            Temp1 = (byte)temp[2] -48;
            Temp2 = (byte)temp[3] -48;
            timeMax = Temp1*10 + Temp2;
            //timeMax=19;
            lcd.print("TERSIMPAN!      ");
        }
        else
        {
            lcd.print("TDK TERSIMPAN! ");
        }
        writeSetting(userMax,quotaMax,timeMin,timeMax);
        menuLevel=1;optionSelect=0;
    }
}
Serial.println("MASTER EVENT");
}
```



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 25 dari 30

3. Menghubungkan master dengan komputer dan meng-*upload* program.

Hubungkan master dengan USB pada *port* master kemudian *upload* program.



Gambar 10. Konektor USB ke master

4. Memprogram Slave/Arduino Nano dengan Arduino Ide

Buka Arduino IDE kemudian isikan program di bawah ini:

```
#include <SPI.h> //library komunikasi SPI
#include <MFRC522.h> // library modul RFID-RC522
#include <Ultrasonic.h> //library modul sensor ultrasonik

//inisialisasi ultrasonik pada pin 4 sebagai TRIG dan 3 sebagai ECHO
Ultrasonic ultrasonic(4,3); // (Trig PIN,Echo PIN)

//pendefinisian untuk komunikasi SPI pada modul RFID-RC522
#define SS_PIN 10
#define RST_PIN 9

//pendefinisian pin untuk mengontrol motor driv
#define PIN_MOTOR 5

MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class

MFRC522::MIFARE_Key key;

// Init array that will store new NUID
byte nuidPICC[4];

String str1="";
bool waitMasterRespons=0;
void gateOpen()
{
    //definisikan kondisi saat gerbang terbuka
    digitalWrite(PIN_MOTOR, 1);
}
void gateClose()
{
    //definisikan kondisi saat gerbang tertutup
    digitalWrite(PIN_MOTOR, 0);
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 26 dari 30

```
}

void setup()
{
    pinMode(PIN_MOTOR, OUTPUT);
    //inisialisasi komunikasi UART dengan server pada baudrate 9600
    Serial.begin(9600);
    //inisialisasi komunikasi SPI
    SPI.begin();
    //inisialisasi modul RFID-RC522
    rfid.PCD_Init();
    for (byte i = 0; i < 6; i++) {
        key.keyByte[i] = 0xFF;
    }
    //mengirim sinyal ke master bahwa slave kondisi siap
    Serial.println("SLAVEREADY");
}

void loop()
{
    if(waitMasterRespons) //menunggu respon dari master
    {
        if(Serial.available())
        {
            str1=Serial.readStringUntil('\n');
            str1=str1.substring(0,str1.length()-1); //menghapus karakter end-line
        }
        if(str1=="TRUE") { trueCard(); waitMasterRespons=0; }
        else if(str1=="FALSE") { falseCard(); waitMasterRespons=0; }
        else if(str1=="MASTER") { masterCard(); waitMasterRespons=0; }
        str1="";
    }
    else
    {
        if(Serial.available())
        {
            str1=Serial.readStringUntil('\n');
            str1=str1.substring(0,str1.length()-1); //menghapus karakter end-line
            if(str1=="MASTER") masterCard();
            str1="";
        }
        readRFID();
    }
}

void readRFID()
{
    //fungsi pembacaan RFID
    // cek pembacaan kartu baru
    if (rfid.PICC_IsNewCardPresent())
    {
        // verifikasi ID sudah terbaca
        if (rfid.PICC_ReadCardSerial())
        {
            // konversi ID ke array
            for (byte i = 0; i < 4; i++)
            {
                nuidPICC[i] = rfid.uid.uidByte[i];
            }
        }
    }
}
```



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04 Revisi: 01 Tgl: April 2019 Hal: 27 dari 30

```
}

//print hasil pembacaan
printHex(rfid.uid.uidByte, rfid.uid.size);
//mengirim sinyal selesai ke modul RFID
rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();

waitMasterRespons=1;
//KETIKA SUDAH MEMBACA KEMUDIAN MENUNGGU JAWABAN MASTER
}
}

void printHex(byte *buffer, byte bufferSize)
{
    //ini adalah fungsi konversi pembacaan RFID ke string untuk dikirim ke master
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? "0" : "");
        Serial.print(buffer[i], HEX);
    }
    Serial.println("");
}

void trueCard()
{
    //ini adalah fungsi jika jawaban master positif/kartu terdaftar.
    bool openGate=1;
    byte range=0;
    byte Step=0;
    unsigned int timeOut=0;
    unsigned int rangeTimeOut=0;
    gateOpen();
    delay(1000);
    while(openGate)
    {
        timeOut++;
        if(timeOut>5000) openGate=0;
        range=ultrasonic.Ranging(CM);
        if(range<50 && Step==0) Step=1;
        while(Step==1)
        {
            range=ultrasonic.Ranging(CM);
            timeOut++;
            if(range>=50) rangeTimeOut++;
            else rangeTimeOut=0;
            if(rangeTimeOut>200 || timeOut>5000)
            {
                openGate=0;
                Step=0;
            }
            delay(1);
        }
        delay(1);
    }
    gateClose();
    //setelah selesai, slave mengirim sinyal siap ke master
    Serial.println("SLAVEREADY");
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

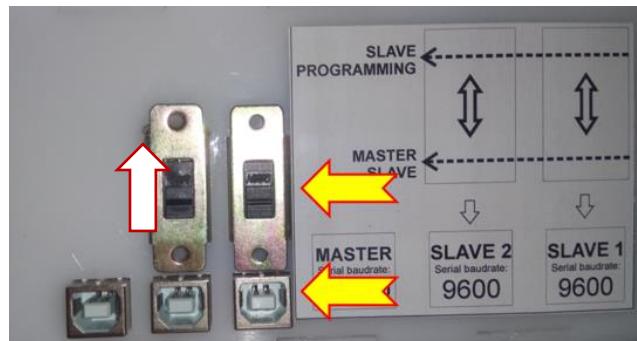
Tgl: April 2019

Hal: 28 dari 30

```
void falseCard()
{
    //ini adalah fungsi jika jawaban master positif/kartu terdaftar.
    gateClose();
    delay(2500);
    //setelah selesai, slave mengirim sinyal siap ke master
    Serial.println("SLAVEREADY");
}
void masterCard()
{
    //ini adalah fungsi untuk kondisi khusus dari master.
    bool master=1;
    while(master)
    {
        gateOpen();
        if(Serial.available())
        {
            str1=Serial.readStringUntil('\n');
            str1=str1.substring(0,str1.length()-1); //remove end character
            if(str1=="FALSE")
            {
                gateClose();
                master=0;
            }
            str1="";
        }
    }
    Serial.println("SLAVEREADY");
}
```

5. Menghubungkan master dengan komputer dan meng-*upload* program.

Hubungkan *slave* dengan USB pada *port* berikut. Kemudian pindah *switch* ke *slave programming* kemudian *upload*.



Gambar 11. Konektor USB dan *switch mode* pada *slave*

6. Analisa pengendalian *slave*.

Pada salah *slave1* / *slave* masuk, pindah *switch* ke *slave programming*.

Hubungkan ke serial monitor pada *baudrate* 9600. Kemudian lakukan aksi berikut secara berurutan.



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

RANCANG-BANGUN SISTEM
PENGATURAN PARKIR
BERBASIS RFID

4x45 menit

No. 04

Revisi: 01

Tgl: April 2019

Hal: 29 dari 30

No	Aksi	Output Serial Monitor	Posisi Portal
1	RFID ditempelkan		
2	Kirim serial: “TRUEX”		
3	Gerakkan tangan di depan sensor ultrasonik		
4	RFID ditempelkan		
5	Kirim serial: “FALSEX”		
6	Gerakkan tangan di depan sensor ultrasonik		
7	RFID ditempelkan		
8	Kirim serial: “MASTERX”		
9	Kirim serial: “FALSEX”		

7. Uji coba Sistem Pengaturan Parkir

Lepas semua kabel USB dan pindah semua posisi *switch slave* menjadi *master-slave*. Analisa respons waktu terhadap RFID masuk pada tabel di bawah ini. Minta tolonglah kepada rekan Anda untuk membantu Anda menghitung waktu menggunakan *stopwatch*.



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4	RANCANG-BANGUN SISTEM PENGATURAN PARKIR BERBASIS RFID	4x45 menit	
No. 04	Revisi: 01	Tgl: April 2019	Hal: 30 dari 30

No	Tap Kartu RFID	Waktu (dalam detik) yang dibutuhkan dari tap kartu hingga ada aksi (portal terbuka atau muncul informasi dari LCD)
1	Terdaftar (1)	
2	Terdaftar (2)	
3	Terdaftar (3)	
4	Master (1)	
5	Master (2)	
6	Tidak terdaftar (1)	
7	Tidak terdaftar (2)	
8	Tidak terdaftar (3)	

Analisa respons waktu terhadap RFID keluar pada tabel di bawah ini.

Gunakan 3 kartu yang sudah di tap di RFID masuk. Minta tolonglah kepada rekan Anda untuk membantu Anda menghitung waktu menggunakan *stopwatch*.

No	Tap Kartu RFID	Waktu (dalam detik) yang dibutuhkan dari tap kartu hingga ada aksi (portal terbuka atau muncul informasi dari LCD)
1	Sudah masuk (1)	
2	Sudah masuk (2)	
3	Sudah masuk (3)	
6	Belum masuk (1)	
7	Belum masuk (2)	
8	Belum masuk (3)	

F. Bahan diskusi

Gunakan kartu master untuk mengubah nilai kuota maksimal dan kapasitas maksimal. Lakukan simulasi keluar-masuk parkir hingga kuota maksimal tercapai dan kapasitas penuh. Buatlah analisis terhadap:

1. isi *file* maupun folder pada SDCard
2. cara master mengendalikan *slave* untuk membuka atau tetap menutup portal.
3. penyebab lebih cepat/lebih lambat rata-rata waktu RFID masuk atau keluar!

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit
	No. 03	Revisi: 01	Tgl: April 2019 Hal: 1 dari 10

A. Tujuan

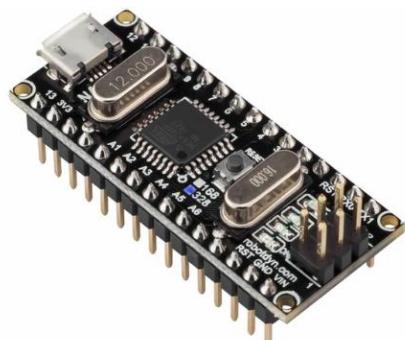
1. Mengetahui konfigurasi Modul RFID-RC522
2. Mengetahui dasar pemrograman Modul RFID-RC522 dengan Arduino
3. Mengetahui cara membaca data identitas dari Modul RFID-RC522

B. Dasar Teori

1. Arduino Nano

Arduino adalah sebuah perangkat keras elektronik dari sebuah platform yang bersifat *open source* dengan *software* yang mudah digunakan yaitu Arduino *Integrated Development Environment* (IDE). Terdapat berbagai jenis Arduino seperti Arduino nano, Arduino uno, Arduino fiod dan masih banyak lagi. Arduino dapat digunakan dalam berbagai jenis perangkat elektronik yang memerlukan pemrograman dan dapat membaca berbagai jenis sensor sehingga banyak dari kalangan teknisi elektronik menggunakan Arduino untuk melengkapi proyek-proyeknya. Selain itu Arduino juga banyak dimanfaatkan sebagai pembelajaran di kalangan universitas.

Arduino Nano adalah modul mikrokontroler yang berbasis pada IC ATmega328 (Arduino Nano 3.0) atau ATmega168 (Arduino Nano 2.x). Dia memiliki 14 digital I/O dengan 6 pin di antaranya mendukung PWM, 8 *input* analog, 1 jalur UART, 16 MHz *crystal oscillator*, sebuah koneksi Mini-B USB, konektor ICSP, dan sebuah tombol reset. Dia juga memiliki keseluruhan komponen yang dapat mendukung jalannya mikrokontroler atau sering disebut dengan sistem minimum.



Gambar 1. Arduino Nano
 (Sumber: <https://www.aliexpress.com/>)

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit
	No. 03	Revisi: 01	Tgl: April 2019 Hal: 2 dari 10

2. Modul RFID-RC522

RFID *reader* adalah perangkat pembaca RFID tag. Sama seperti RFID tag, RFID *reader* dapat bersifat aktif dan pasif. RFID *reader* pasif berarti tidak memiliki daya sendiri untuk melakukan pembacaan sehingga harus disuplai dari RFID tag aktif. Sifat tersebut membuat RFID *reader* pasif hanya dapat membaca RFID tag aktif. RFID *reader* aktif memiliki daya sendiri dan dapat memancarkan sinyal interrogator ke RFID tag pasif. Sinyal interrogator menginduksi RFID tag sehingga menjadi sumber daya pada RFID tag pasif tersebut. Pada penelitian ini, modul RFID *reader* yang digunakan adalah modul RFID-RC522 yang berbasis IC MFRC522.

IC MFRC522 yang berada pada modul RFID *reader* sebetulnya dapat menggunakan 3 antar muka komunikasi yaitu *Serial Peripheral Interface* (SPI), I2C, dan UART. Namun pada modul RFID *reader* yang digunakan, komponen yang terpasang hanya mendukung antarmuka komunikasi SPI saja. Modul MFRC522 menggunakan komunikasi SPI karena transfer data dapat terjadi dengan cepat. Modul MFRC522 ini nantinya akan bekerja pada Arduino Nano yang bertindak sebagai *slave*. Informasi identitas yang dimiliki RFID tag kemudian akan dikirim menggunakan antarmuka komunikasi serial UART oleh *slave* ke *master* yaitu Arduino Mega.



Gambar 2. Bentuk Fisik Modul RFID-RC522

(Sumber: <https://www.lazada.com.my/>)



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

MENAKSES MODUL
RFID-RC522 DENGAN
ARDUINO NANO

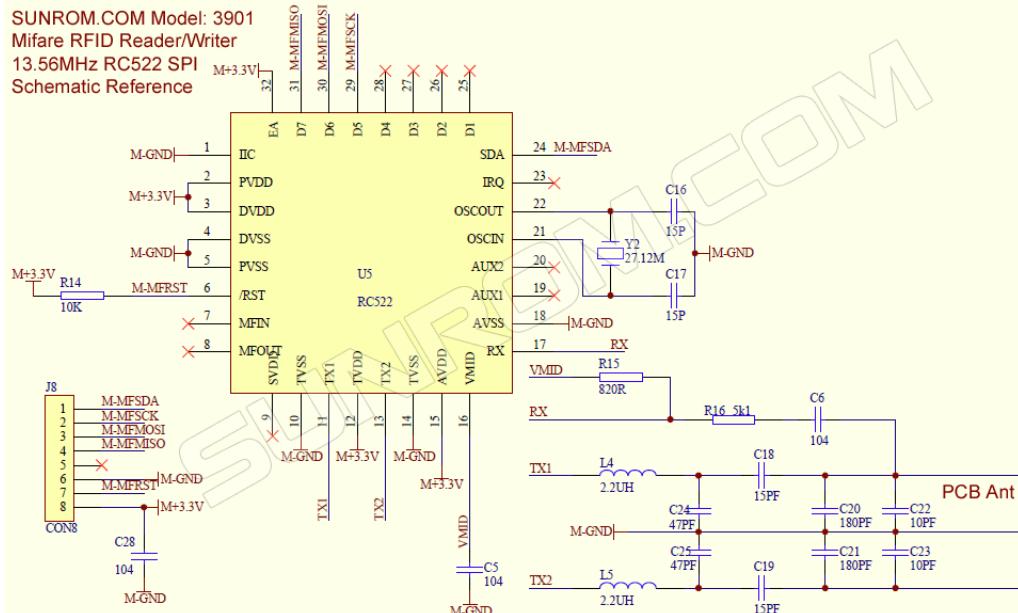
4x45 menit

No. 03

Revisi: 01

Tgl: April 2019

Hal: 3 dari 10



Gambar 3. Skema Modul RFID-RC522

(Sumber: <https://www.sunrom.com/>)

Tabel 1. Fungsi kaki-kaki pada modul RFID-RC522

Pin	Label	Fungsi
1	SDA	Pin <i>Slave Select</i> untuk komunikasi SPI
2	SCK	Pin <i>Serial Clock</i> untuk komunikasi SPI
3	MOSI	Pin <i>MOSI</i> untuk komunikasi SPI
4	MISO	Pin <i>MISO</i> untuk komunikasi SPI
5	IRQ	tidak terhubung
6	GND	<i>Ground</i> atau 0V
7	RST	Pin <i>Reset</i> pada komunikasi SPI
8	3,3V	Tegangan masukan 3,3V

3. Komunikasi SPI

Serial Peripheral Interface (SPI) merupakan komunikasi serial sinkron karena menggunakan jalur terpisah antara data dan *clock* (Suhaeb dkk., 2017:135). Jalur data dan *clock* yang terpisah membuat komunikasi SPI menjadi lebih cepat



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENAKSES MODUL
RFID-RC522 DENGAN
ARDUINO NANO

4x45 menit

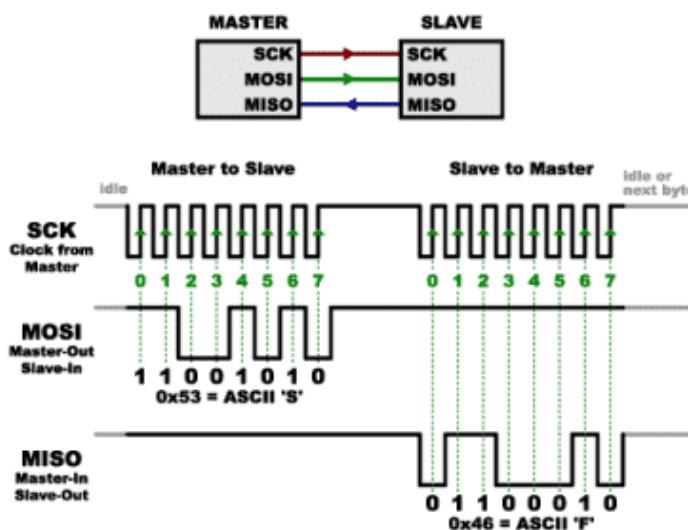
No. 03

Revisi: 01

Tgl: April 2019

Hal: 4 dari 10

karena data pada pengirim dan penerima lebih akurat. Pada umumnya komunikasi menggunakan SPI memiliki beberapa jalur kawat yaitu *Vcc*, *Ground*, *Serial Clock (SCK)*, *Master-Output-Slave-Input (MOSI)*, *Master-Input-Slave-Output (MISO)*, *Slave-Select (SS)* dan beberapa kasus menggunakan pin *Reset (R)*. Namun pada dasarnya SPI hanya terdiri dari 3 pin, yaitu *SCK*, *MOSI*, dan *MISO*.



Gambar 4. Konsep Komunikasi SPI

(Sumber: Suhaeb, dkk., 2017:136)

Contoh komponen yang menggunakan komunikasi SPI sebagai antarmukanya adalah *RFID Reader* yaitu modul MFRC522, modul MicroSD, *downloader* mikrokontroler, dan sebagainya. Kecepatan transfer data pada SPI dapat mencapai 10Mbps.

C. Alat dan Bahan

1. Modul RFID-RC522
2. LED dan resistor 220 Ohm
3. RFID Tag (Kartu RFID)
4. Arduino Nano
5. Kabel USB
6. Kabel *jumper* secukupnya
7. Komputer dengan *Software Arduino IDE* yang sudah terpasang.

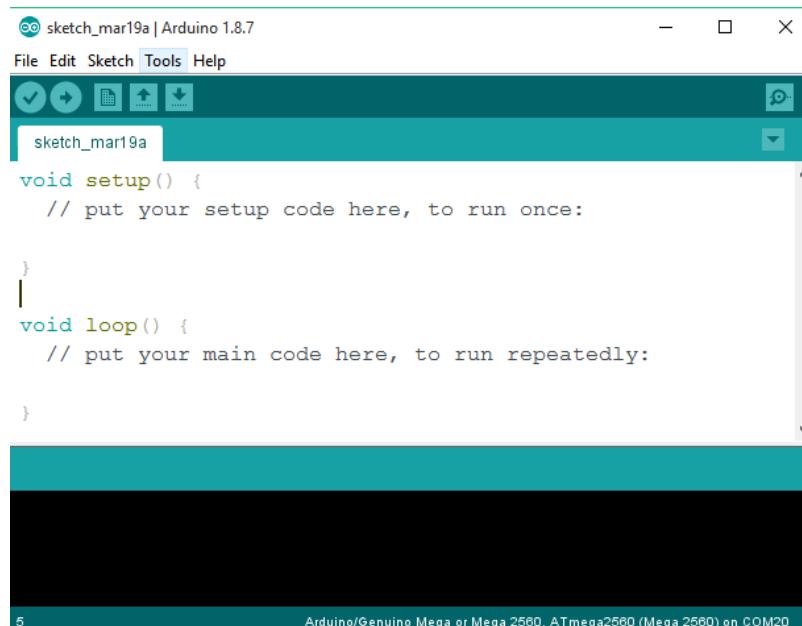
	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit	
No. 03	Revisi: 01	Tgl: April 2019	Hal: 5 dari 10

D. Keselamatan Kerja

1. Pastikan saat memulai membuat rangkaian dalam kondisi *off* atau tanpa tegangan.
2. Gunakan alat dan bahan sesuai dengan fungsinya.
3. Jauhkan alat dan bahan yang tidak digunakan dari meja kerja.
4. Dalam merangkai pastikan sesuai dengan langkah kerja yang sudah ditentukan.
5. Tanyalah kepada pengampu apabila terjadi kesalahan dalam pengoperasian dan apabila terdapat penjelasan atau materi yang kurang dimengerti.

E. Langkah Kerja

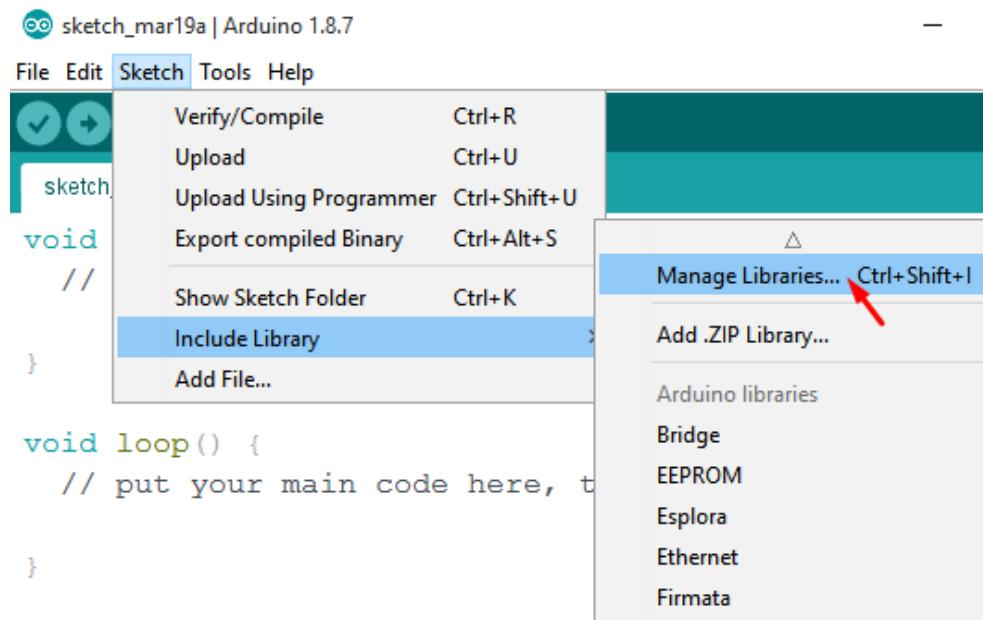
1. Memasang *library* SD pada Arduino IDE
 - a. Buka software Arduino IDE, maka akan muncul tampilan utama.



Gambar 5. Tampilan utama Arduino IDE

- b. Koneksikan Komputer ke internet kemudian pilih "Sketch" – pilih "Include Library" – pilih "Manage Libraries".

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit	
No. 03	Revisi: 01	Tgl: April 2019	Hal: 6 dari 10



Gambar 6. Menampilkan "Manage Libraries"

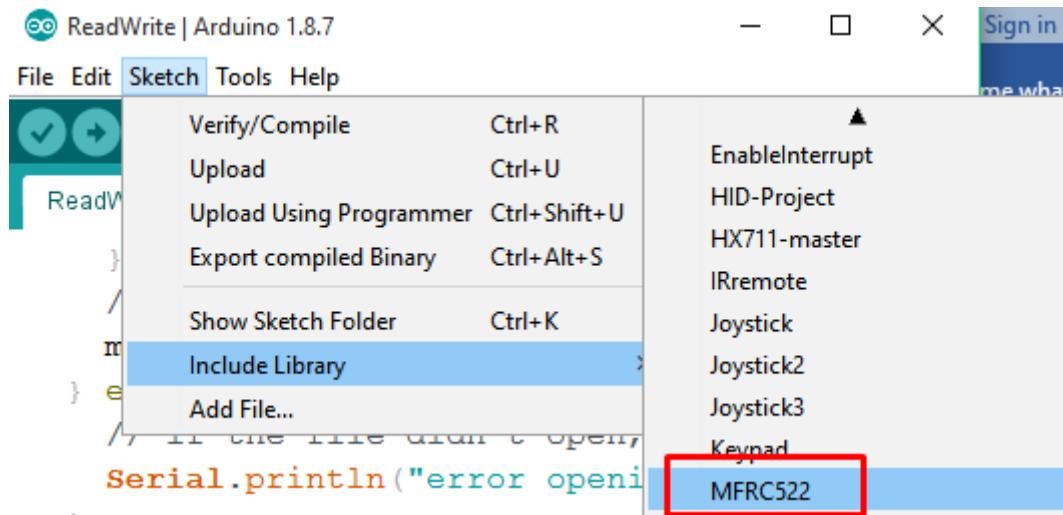
Setelah itu akan muncul jendela “Library Manager” – kemudian ketikan “RC522” dan tekan “Enter” – cari yang di-*publish* oleh “Sparkfun” – Pilih versi terbaru – Klik “Install” – kemudian tunggu proses hingga selesai.



Gambar 7. Tampilan Jendela "Library Manager"

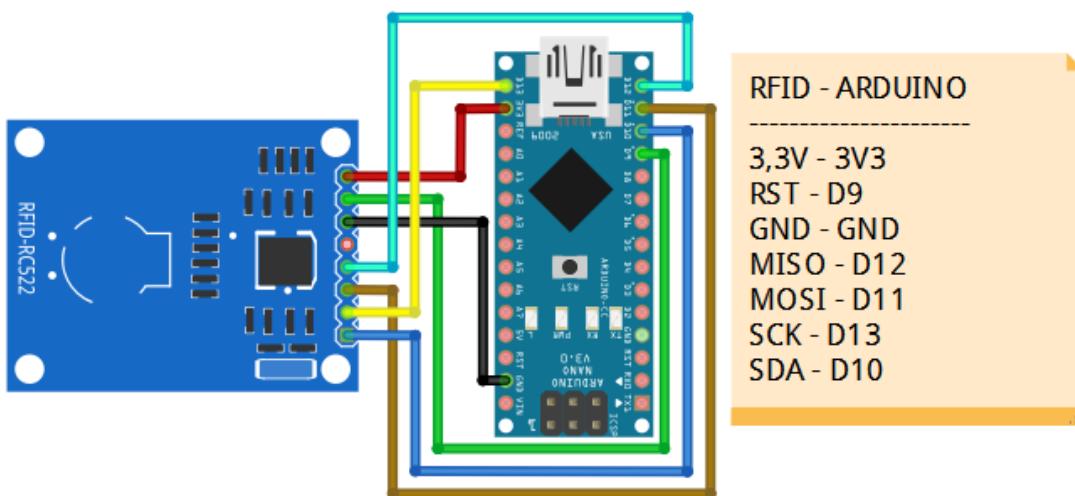
- Untuk menguji apakah *library* MFRC522 sudah masuk atau belum silakan klik “Sketch” – kemudian geser kursor ke “Include Library” – pastikan ada *library* MFRC522 di situ.

	JURUSAN PENDIDIKAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit	
No. 03	Revisi: 01	Tgl: April 2019	Hal: 7 dari 10



Gambar 8. Cek *library*

2. Mengidentifikasi rangkaian Arduino Nano dengan Modul M RFID-RC522
Modul RTC dan Arduino Mega sudah tersambung dan terangkai dalam Sistem Pengaturan Parkir dengan koneksi seperti Gambar 9.



Gambar 9. Skema pengabelan Modul RFID-RC522 dengan Arduino Nano pada Sistem Pengaturan Parkir

3. Menghubungkan Arduino pada Sistem Pengaturan Parkir dengan komputer
Hubungkan kabel USB dengan perangkat dan pindah *switch* dari MASTER-SLAVE menjadi SLAVE PROGRAMMING.



JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

SEMESTER 4

MENAKSES MODUL
RFID-RC522 DENGAN
ARDUINO NANO

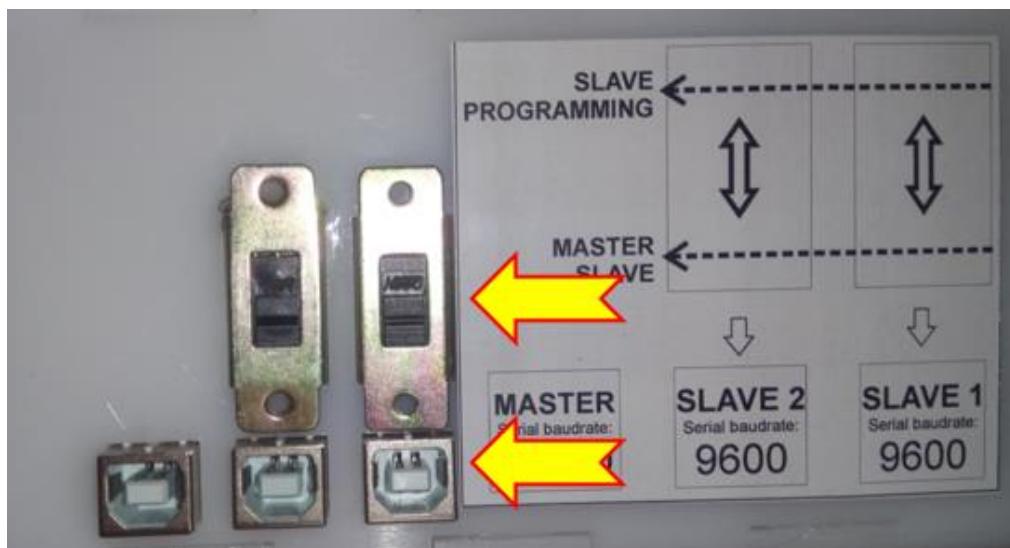
4x45 menit

No. 03

Revisi: 01

Tgl: April 2019

Hal: 8 dari 10



Gambar 10. Konektor USB dan switch mode pada slave

4. Memprogram dengan Arduino Ide.

Buka Arduino IDE kemudian isikan program di bawah ini:

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 rfid(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
// Membuat array yang akan menyimpan pembacaan ID RFID.
byte nuidPICC[4];
void setup() {
  Serial.begin(9600);
  SPI.begin(); // Inisialisasi SPI
  rfid.PCD_Init(); // Inisialisasi MFRC522
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  Serial.println(F("This code scan the MIFARE Classsic NUID."));
  Serial.print(F("Using the following key:"));
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4

MENAKSES MODUL
RFID-RC522 DENGAN
ARDUINO NANO

4x45 menit

No. 03

Revisi: 01

Tgl: April 2019

Hal: 9 dari 10

```
printHex(key.keyByte, MFRC522::MF_KEY_SIZE);  
}  
void loop()  
{  
    readRFID();  
}  
void readRFID()  
{  
    // Membaca kartu  
    if (rfid.PICC_IsNewCardPresent())  
    {  
        // Verifikasi pembacaan ID RFID  
        if (rfid.PICC_ReadCardSerial())  
        {  
            // Menyimpan hasil pembacaan ke array  
            for (byte i = 0; i < 4; i++)  
            {  
                nuidPICC[i] = rfid.uid.uidByte[i];  
            }  
            printHex(rfid.uid.uidByte, rfid.uid.size);  
            // RFID reader standby.  
            rfid.PICC_HaltA();  
            rfid.PCD_StopCrypto1();  
        }  
    }  
}  
void printHex(byte *buffer, byte bufferSize) {  
    for (byte i = 0; i < bufferSize; i++) {  
        Serial.print(buffer[i] < 0x10 ? "0" : "");  
        Serial.print(buffer[i], HEX);  
    }  
}
```



**JURUSAN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

SEMESTER 4	MENAKSES MODUL RFID-RC522 DENGAN ARDUINO NANO	4x45 menit	
No. 03	Revisi: 01	Tgl: April 2019	Hal: 10 dari 10

5. *Upload* program ke perangkat.
6. Jalankan serial monitor melalui Arduino IDE.
7. Kirim perintah ke Arduino melalui serial monitor kemudian lihat hasilnya dan masukkan pada tabel di bawah ini:

No	Tempelkan kartu RFID	<i>Output Serial Monitor</i>
1	Kartu ke 1	
2	Kartu ke 2	
3	Kartu ke 3	
4	Kartu ke 4	
5	Kartu ke 5	

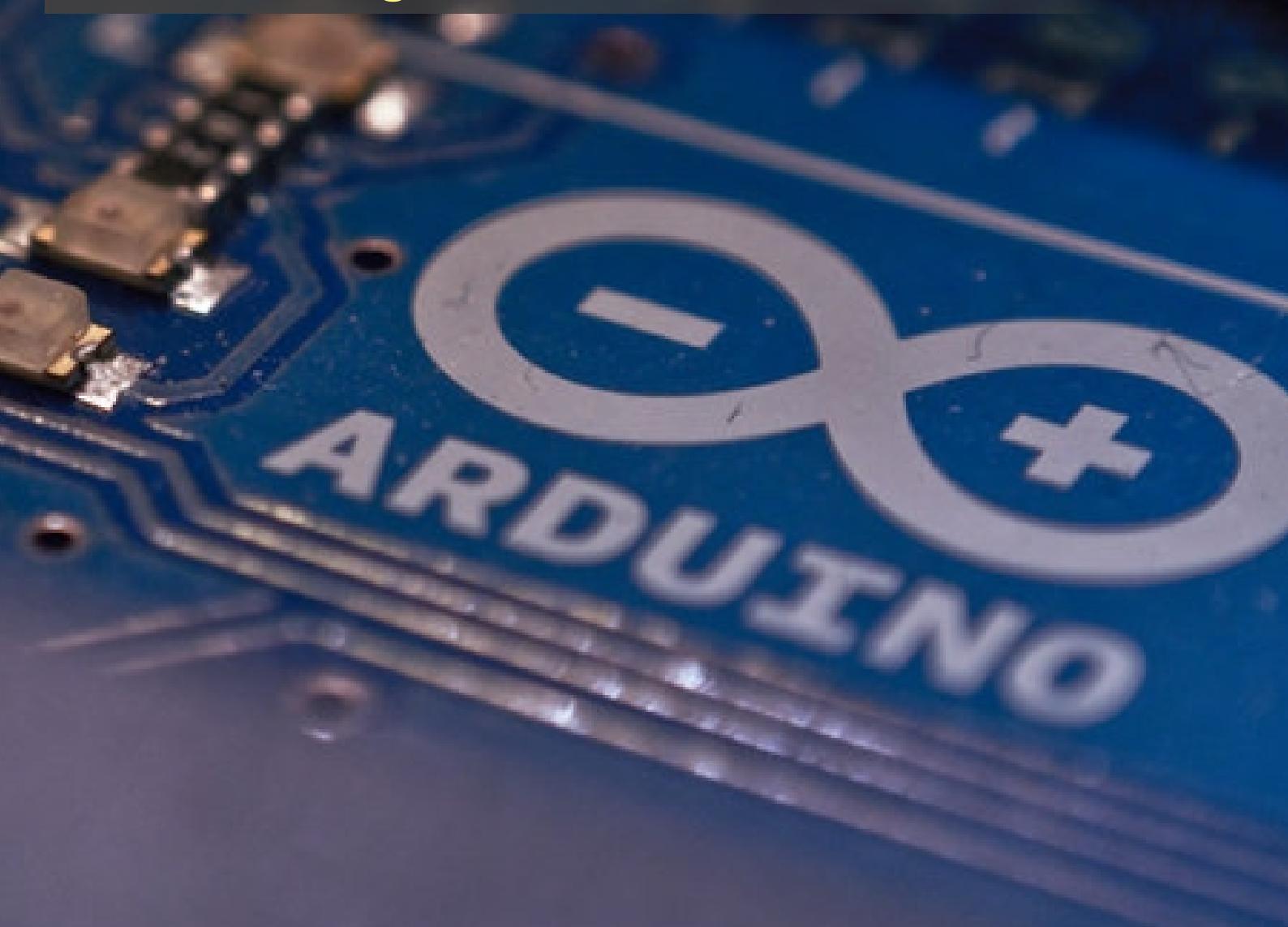
F. Bahan Diskusi

Hubungkan *port Portal Masuk* dan *Main Controller* Sistem Pengaturan Parkir Arduino Nano (*slave*) RFID masuk. Untuk membuka portal cukup memberi sinyal HIGH (1), dan untuk menutup portal cukup memberi sinyal LOW (0). Pin portal terhubung pada PIN nomor 5 Arduino Nano. Buatlah program Arduino yang dapat membuka portal hanya ketika kartu ke 1 dan 2 hidup atau ketika ada perintah “TRUE” dari *serial monitor*!

Lampiran 13. Modul Pembelajaran

MODUL MIKROKONTROLER

- *Arduino*
- *Radio Frequency Identification (RFID)*
- *Real Time Clock DS3231*
- *Micro SD Card*
- Sistem Pengaturan Parkir



Isnan Nabawi - 15518241002
Pendidikan Teknik Mekatronika
FT UNY 2019

**SISTEM PENGATURAN PARKIR BERBASIS
RADIO FREQUENCY IDENTIFICATION (RFID)**

Kata Pengantar

Puji syukur kita ucapkan ke hadirat Allah SWT yang telah melimpahkan karunia, rahmat, serta petunjuk-Nya, sehingga penyusun dapat menyelesaikan modul praktikum Sistem Pengaturan Parkir berbasis *Radio Frequency Identification* (RFID). Modul ini disusun agar dapat membantu guru dan siswa dalam kegiatan pembelajaran praktik mikrokontroler pada mata kuliah praktik mikrokontroler.

Modul praktikum ini memuat materi pembelajaran praktik mikrokontroler untuk sistem pengaturan parkir berbasis RFID. Komponen yang ada pada media pembelajaran sistem pengaturan parkir dibahas dalam modul ini. Komponen tersebut adalah komponen-komponen utama yang ada pada sistem seperti mikrokontroler Arduino, RFID, *Real Time Clock* (RTC), MicroSD, dan Motor DC. Materi yang dibahas termasuk pembahasan mendasar tentang komponen maupun cara memprogram dan pengendaliannya.

Tentu tidak ada hal yang sempurna di dunia ini, semoga modul praktikum ini bermanfaat bagi dunia pendidikan untuk meningkatkan kompetensi peserta didik khususnya di lingkungan Universitas Negeri Yogyakarta. Modul praktikum ini masih banyak kekurangan, saya mengharapkan saran dan kritik dari pengguna modul praktikum ini untuk perbaikan di masa yang akan datang. Penyusun mengucapkan terima kasih kepada semua pihak yang turut membantu dalam penyusunan modul praktikum ini.

Penyusun

Daftar Isi

COVER.....	I
KATA PENGANTAR	II
DAFTAR ISI	III
DAFTAR TABEL	IV
DAFTAR GAMBAR	V
BAGIAN I – MIKROKONTROLLER ARDUINO	1
A. ARDUINO	1
B. PROTOKOL KOMUNIKASI ARDUINO	3
C. PEMROGRAMAN DASAR ARDUINO	6
BAGIAN II – RADIO FREQUENCY IDENTIFICATION (RFID)	14
A. MODUL RFID RC-522.....	14
B. PEMROGRAMAN MODUL RFID DENGAN ARDUINO.....	18
BAGIAN III – REAL TIME CLOCK (RTC)	20
A. MODUL RTC DS3231	20
B. PEMROGRAMAN RTC DENGAN ARDUINO	21
BAGIAN IV – MODUL MICRO SD CARD	25
A. MODUL MICROSD CARD <i>READER-WRITER</i>	25
B. PEMROGRAMAN MODUL MICRO SD DENGAN ARDUINO	26
BAGIAN V – MOTOR DC.....	29
A. <i>BRUSHED</i> MOTOR DC.....	29
B. MENGENDALIKAN <i>BRUSHED</i> MOTOR DC.....	30
BAGIAN VI – SISTEM PENGATURAN PARKIR BERBASIS RFID	33
A. PENGENALAN SISTEM.....	33
B. BAGAN DAN <i>FLOWCHART</i> SISTEM.....	35
C. BAGIAN-BAGIAN UNIT	37
D. PERAKITAN.....	38
E. PENGOPERASIAN	39
DAFTAR PUSTAKA	41
LAMPIRAN	42

Daftar Tabel

Tabel 1. Spesifikasi Arduino Mega 2560.....	2
Tabel 2. Spesifikasi Arduino Nano	3
Tabel 3. Beberapa tipe data pada bahasa C	9
Tabel 4. Beberapa instruksi dalam bahasa C	11
Tabel 5. Daftar Alokasi Frekuensi RFID	15
Tabel 6. Fungsi kaki-kaki pada modul RFID RC522	18
Tabel 7. Fungsi tiap pin modul RTC DS3231	20
Tabel 8. Fungsi untuk mengambil nilai waktu dari RTC.....	22
Tabel 9. Fungsi untuk mengatur nilai waktu pada RTC	22
Tabel 10. Fungsi kaki-kaki pada modul <i>micro sd card</i> adapter.....	25
Tabel 11. Fungsi untuk mengakses micro SD.....	26
Tabel 12. Menu pada <i>master setting</i>	40

Daftar Gambar

Gambar 1. Arduino Mega	1
Gambar 2. Arduino Nano	2
Gambar 3. Konsep Komunikasi SPI	5
Gambar 4. Start dan Stop bit pada I2C	5
Gambar 5. Skema komunikasi multi-slave pada I2C.....	6
Gambar 6. Modul RFID <i>reader</i> MFRC 522	14
Gambar 7. Ilustrasi RFID Tag.....	16
Gambar 8. Skema Modul RFID <i>reader</i> MFRC522	17
Gambar 9. Modul <i>Real Time Clock</i> (RTC)	20
Gambar 10. Skema modul RTC DS3231	21
Gambar 11. Modul SD <i>Card Adapter</i>	25
Gambar 12. Motor DC dengan <i>gearbox</i>	29
Gambar 13. Visualisasi cara kerja motor DC.....	30
Gambar 14. Rangkaian H-Bridge dengan 4 transistor	31
Gambar 15. Desain Posisi <i>Main Controller</i> dengan Sistem Portal.....	34
Gambar 16. Bagan Sistem.....	35
Gambar 17. <i>Flowchart</i> saat pengguna keluar	35
Gambar 18. <i>Flowchart</i> saat pengguna masuk	36
Gambar 19. Bentuk fisik <i>main controller</i> dengan keterangan	37
Gambar 20. Bentuk fisik bagian portal dengan keterangan	37
Gambar 21. Kabel konektor dan Adaptor	38
Gambar 22. Rekomendasi posisi unit	39

BAGIAN I – MIKROKONTROLLER ARDUINO

A. Arduino

Arduino adalah sebuah platform elektronik yang bersifat *open source* serta mudah digunakan. Hal tersebut ditujukan agar siapa pun dapat membuat proyek interaktif dengan mudah dan menarik (Wicaksono & Hidayat, 2017:1). Dalam penelitian ini, Arduino yang digunakan adalah Arduino Mega 2560 dan Arduino Nano. Arduino Mega 2560 digunakan sebagai *master*. Sedangkan Arduino Nano digunakan sebagai *slave*.

Arduino Mega 2560 adalah modul mikrokontroler yang berbasis pada IC ATmega2560. Dia memiliki 54 digital *input/output* (I/O) dengan 14 pin di antaranya mendukung *pulse-width modulation* (PWM), 16 *input* analog, 4 jalur *Universal Asynchronous Receiver-Transmitter* (UART), 16 MHz *crystal oscillator*, sebuah koneksi *universal serial bus* (USB), konektor daya, konektor *in circuit serial programming* (ICSP), dan sebuah tombol reset. Dia juga memiliki keseluruhan komponen yang dapat mendukung jalannya mikrokontroler atau sering disebut dengan sistem minimum. Spesifikasi Arduino Mega 2560 dapat dilihat pada Tabel 1.



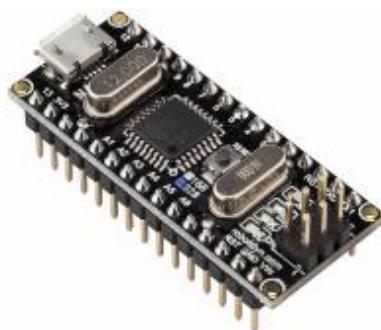
Gambar 1. Arduino Mega

(Sumber: <https://store.arduino.cc/>)

Tabel 1. Spesifikasi Arduino Mega 2560

Mikrokontroler	ATmega2560
Tegangan kerja	5 V
Tegangan masukan (rekomendasi)	7-12 V
Tegangan masukan (maksimal)	6-20 V
Digital I/O	54 (14 PWM)
Analog <i>input</i> (ADC)	16
Arus maksimal per digital I/O	40 mA
Arus maksimal pada pin 3,3V	50 mA
Memori <i>flash</i>	256 KB (8KB sudah digunakan untuk <i>bootloader</i>)
<i>Static Random Access Memory</i> (SRAM)	8 KB
<i>Electrically Erasable Programmable Read-Only Memory</i> (EEPROM)	4 KB
<i>Clock Speed</i>	16 MHz

Arduino Nano adalah modul mikrokontroler yang berbasis pada IC ATmega328 (Arduino Nano 3.0) atau ATmega168 (Arduino Nano 2.x). Dia memiliki 14 digital I/O dengan 6 pin di antaranya mendukung PWM, 8 *input* analog, 1 jalur UART, 16 MHz *crystal oscillator*, sebuah koneksi Mini-B USB, konektor ICSP, dan sebuah tombol reset. Dia juga memiliki keseluruhan komponen yang dapat mendukung jalannya mikrokontroler atau sering disebut dengan sistem minimum. Spesifikasi Arduino Nano dapat dilihat pada Tabel 2.



Gambar 2. Arduino Nano
(Sumber: <https://www.aliexpress.com/>)

Tabel 2. Spesifikasi Arduino Nano

Mikrokontroler	ATmega328 atau ATmega168
Tegangan kerja	5 V
Tegangan masukan (rekomendasi)	7-12 V
Tegangan masukan (maksimal)	6-20 V
Digital I/O	14 (6 PWM)
Analog <i>input</i> (ADC)	8
Arus maksimal per digital I/O	40 mA
Arus maksimal pada pin 3,3V	50 mA
Memori <i>flash</i>	16 KB (ATmega168) atau 32 KB (ATmega328) dengan 2 KB sudah digunakan untuk <i>bootloader</i>)
<i>Static Random Access Memory</i> (SRAM)	1 KB (ATmega168) atau 2 KB (ATmega328)
<i>Electrically Erasable Programmable Read-Only Memory</i> (EEPROM)	512 bytes (ATmega168) atau 1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz

B. Protokol Komunikasi Arduino

1. Komunikasi Serial Asinkron *UART*

Komunikasi Serial Asinkron *Universal Asynchronous Receiver and Transmitter* (UART) merupakan alternatif yang lebih murah menggantikan komunikasi paralel. Transfer data pada komunikasi paralel memerlukan 8 konduktor untuk mentransfer 8 bit data secara langsung sehingga lebih boros biaya pada pengadaan *hardware* (Suhaeb dkk., 2017:118). Komunikasi serial UART menawarkan biaya yang lebih murah karena hanya menggunakan 4 jalur kawat saja, yaitu Vcc, Ground, Tx, dan Rx. Komunikasi serial yang umum digunakan adalah standar komunikasi serial RS-232.

Arduino Mega menyediakan jalur komunikasi serial sebanyak 4 jalur yaitu Tx-Rx pada pin 0-1, 14-15, 16-17, 18-19, dan 20-21. Sedangkan Arduino Nano menyediakan jalur komunikasi serial sebanyak 1 jalur yaitu pin Tx-Rx pada pin 0

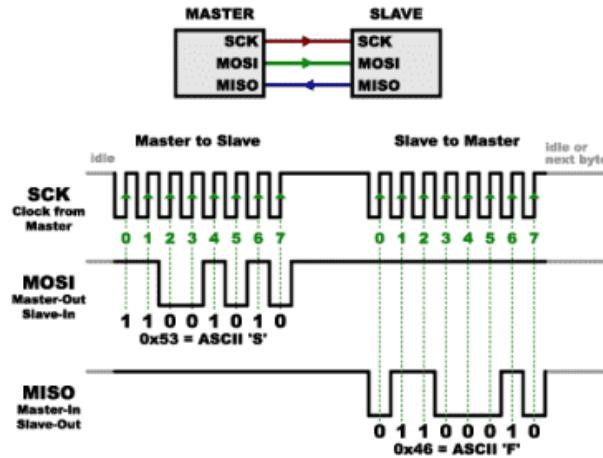
dan 1. Sehingga pada sistem *master-slave*, Arduino Mega lebih cocok digunakan sebagai *master* dan Arduino Nano cocok digunakan sebagai *slave*.

Komunikasi asinkron membutuhkan suatu pengaturan agar data yang dikirim dan diterima bisa sama. Pengaturan yang dimaksud adalah kecepatan transmisi pada pengirim dan penerima yang harus sama. Kecepatan transmisi ini disebut juga dengan *baud rate*. Pada pemrograman serial asinkron pada Arduino, pengaturan kecepatan transmisi dilakukan dengan perintah *Serial.begin(9600)* jika menggunakan *baud rate* sebesar 9600. Pengaturan kecepatan transmisi tersebut harus dilakukan pada master maupun *slave*.

2. *Serial Peripheral Interface (SPI)*

Serial Peripheral Interface (SPI) merupakan komunikasi serial sinkron karena menggunakan jalur terpisah antara data dan *clock* (Suhaeb dkk., 2017:135). Jalur data dan *clock* yang terpisah membuat komunikasi SPI menjadi lebih cepat karena data pada pengirim dan penerima lebih akurat. Pada umumnya komunikasi menggunakan SPI memiliki beberapa jalur kawat yaitu *Vcc*, *Ground*, *Serial Clock* (SCK), *Master-Output-Slave-Input* (MOSI), *Master-Input-Slave-Output* (MISO), *Slave-Select* (SS) dan beberapa kasus menggunakan pin *Reset* (R). Namun pada dasarnya SPI hanya terdiri dari 3 pin, yaitu SCK, MOSI, dan MISO.

Contoh komponen yang menggunakan komunikasi SPI sebagai antarmukanya adalah *RFID Reader* yaitu modul MFRC522, modul MicroSD, *downloader* mikrokontroler, dan sebagainya. Kecepatan transfer data pada SPI dapat mencapai 10Mbps.

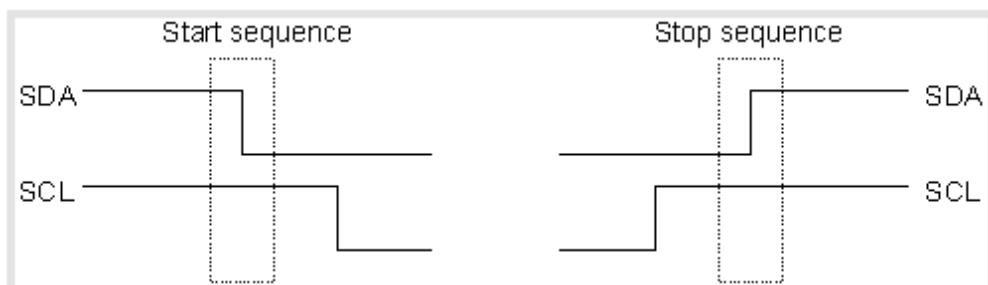


Gambar 3. Konsep Komunikasi SPI

(Sumber: Suhaeb, dkk., 2017:136)

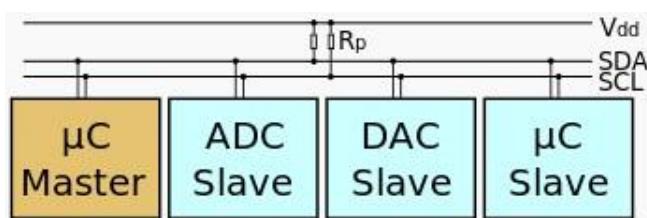
3. *Inter Integrated Circuit* (I2C)

Inter Integrated Circuit (I2C) adalah standar komunikasi serial dua arah yang menggunakan dua saluran yang didesain untuk mengirim maupun menerima data (Suhaeb, dkk., 2017: 154). Komunikasi pada I2C menggunakan 2 pin yaitu *Serial Clock* (SCL) dan *Serial Data* (SDA) yang membawa informasi data antara I2C dengan pengontrolnya. Sinyal *start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 1 menjadi 0 pada saat SCL 1. Sinyal *stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 0 menjadi 1 pada saat SCL 1.



Gambar 4. Start dan Stop bit pada I2C
(Sumber: <https://proyekarduino.wordpress.com/>)

Meski hanya memiliki 2 pin komunikasi, I2C memiliki kemampuan untuk digunakan *multi-slave* karena I2C memiliki sistem pengalamatan. Sehingga jika dimiliki 1 *master* dan 3 *slave*, pin yang digunakan pada master hanya 2 pin (SDA dan SCL) saja. Contoh komponen yang menggunakan komunikasi serial I2C adalah RTC. Pemrograman Komunikasi Serial I2C pada Arduino menggunakan *library* "Wire.h".



Gambar 5. Skema komunikasi multi-slave pada I2C
(Sumber: <https://proyekarduino.wordpress.com/>)

C. Pemrograman Dasar Arduino

Arduino menggunakan bahasa C dalam pemrogramannya. Saat Anda membuka program Arduino IDE pertama kali, Anda akan mendapatkan sebuah program *default* yang menampilkan "void setup();" dan "void loop();" Semua kode program yang ada dalam **void setup** akan dibaca sekali oleh Arduino. Biasanya isinya berupa kode perintah untuk menentukan fungsi pada sebuah pin. Contoh kodennya seperti:

```
pinMode(13, OUTPUT); // menentukan pin 13 sebagai OUTPUT  
pinMode(3, INPUT); // menentukan pin 3 sebagai INPUT
```

Adapun untuk komunikasi antara Arduino dengan komputer menggunakan:

```
Serial.begin(9600); //Inisialisasi komunikasi serial UART
```

Semua kode program yang ada di **void loop** akan dibaca setelah void setup dan akan dibaca terus menerus oleh Arduino. Isinya berupa kode-kode perintah kepada pin INPUT dan OUTPUT pada Arduino. Contoh:

```
digitalWrite(13, HIGH); //memberi sinyal HIGH(5V) pada pin 13  
digitalWrite(13, LOW); //memberi sinyal LOW(5V) pada pin 13
```

Adapun untuk mengirim nilai tertentu pada sebuah sensor di Serial Monitor, bisa menggunakan:

```
Serial.print(NILAI_TERTENTU); //mengirim nilai tertentu via  
Serial UART.
```

Untuk mengirim string/teks, bisa menggunakan:

```
Serial.print("Selamat Datang"); //menampilkan teks Selamat  
Datang pada Serial Monitor
```

Komunikasi Serial UART dapat diakses langsung pada komputer yaitu dengan membuka Serial Monitor pada Arduino IDE. Untuk membuka Serial Monitor sendiri pada Arduino, pilih menu “Tools” kemudian pilih Serial Monitor atau dengan menekan kombinasi **CTRL+SHIFT+M** di *keyboard*. Serial Monitor bisa

jugaberikan ikon kaca pembesar () di Arduino.

CATATAN PADA PROGRAM

Anda bisa membuat catatan pada program dan tidak akan dibaca oleh Arduino, dengan cara mengetikan **//** kemudian mengetikan catatannya, seperti:

```
void loop() {  
    // catatan pada baris ini tidak akan dibaca oleh program  
}
```

Tapi pemakaian tanda // hanya berfungsi untuk catatan satu baris saja, jika Anda ingin membuat catatan yang panjang yaitu berupa paragraf. Maka pertama Anda ketikan /* lalu ketikan catatan Anda, dan jika sudah selesai tutup dengan kode */ .

Contoh:

```
void loop() {  
    /* apapun yang Anda mau ketikan disini tidak  
     * akan dibaca oleh program  
     * sepanjang apapun Anda mengetiknya  
    */  
}
```

KURUNG KURAWAL {}

Digunakan untuk menentukan awal dan akhir dari program. Karena seperti bahasa pemrograman pada umumnya, Arduino membaca mulai dari atas hingga ke bawah.

```
void loop()  
{  
    ....program  
    ....program
```

```
....program  
}
```

TITIK KOMA ;

Setiap baris kode pada Arduino harus diakhiri dengan tanda titik koma (;)

```
void setup(){  
pinMode(13, OUTPUT);  
}  
  
void loop(){  
digitalWrite(13, HIGH);  
}
```

VARIABEL

Variabel adalah kode program yang digunakan untuk menyimpan suatu nilai pada sebuah nama. Variabel memiliki beberapa tipe data. Setiap tipe data memiliki size dan jangkauan nilai yang berbeda. Semakin banyak *size* variabel semakin banyak juga size program yang akan di *upload* ke mikrokontroler. *Programmer* harus jeli dalam memilih tipe data untuk variabelnya agar program yang dibuat lebih efisien. Berikut adalah tabel tipe-tipe variabel yang sering digunakan pada pemrograman Arduino.

Tabel 3. Beberapa tipe data pada bahasa C

Type	Size(bits)	Range(jangkauan)
bool	8	0 dan 1
char	8	-128 sampai 127
unsigned char	8	0 sampai 255

signed char	8	-128 sampai 127
byte	8	0 sampai 255
Int	16	-32768 sampai 32767
short int	16	-32768 sampai 32767
unsigned int	16	0 sampai 65535
signed unt	16	-32768 sampai 326767
long int	32	-2147483648 sampai 2147483648
unsigned long int	32	0 sampai 4294967295
signed long int	32	-2147483648 sampai 2147483648
Float	32	$\pm 1.145e-38$ sampai $3.402e38$
Double	32	$\pm 1.145e-38$ sampai $3.402e38$

OPERATOR MATEMATIKA

Operator matematika digunakan untuk memanipulasi nilai dengan perhitungan matematika sederhana seperti: penjumlahan, pengurangan, sama dengan, dan sebagainya.

= (sama dengan) (contoh **x=10*2** (x sekarang jadi **20**))
% (persentase) (contoh **12%10** (hasilnya yaitu **2**))
+ (penambahan)
- (pengurangan)
* (perkalian)
/ (pembagian)

OPERATOR PERBANDINGAN KONDISI

Digunakan untuk melakukan perbandingan secara logika.

== (sama dengan) contoh: **15 == 10** **FALSE** atau **15 == 15** **TRUE**

!= (tidak sama dengan) contoh: **15 != 10** **TRUE** atau **15 != 15** **FALSE**

< (lebih kecil dari) contoh: **15 < 10** **FALSE** atau **12 < 14** **TRUE**

> (lebih besar dari) contoh: **15 > 19** **TRUE** atau **15 > 10** **FALSE**

INSTRUKSI PENGENDALI

Instruksi pengendali adalah sebuah instruksi yang akan berjalan ketika pada kondisi tertentu. Berikut adalah instruksi-instruksi dalam bahasa C yang sering digunakan:

Tabel 4. Beberapa instruksi dalam bahasa C

No.	Fungsi	Bahasa Pemrograman
1.	Syarat	<pre>if (kondisi) { (aksi yang dikerjakan) };</pre>
2.	Percabangan	<pre>if (kondisi) { (aksi yang dikerjakan) } else if (kondisi) { (aksi yang dikerjakan) } else { (aksi yang dikerjakan) };</pre>
3.	Percabangan	<pre>switch (variable) { case nilai_variabel_ke-1 { (aksi yang dikerjakan) } case nilai_variabel_ke-2 { (aksi yang dikerjakan) } default: { (aksi yang dikerjakan) } }</pre>
4.	Melompat	<pre>goto alamat_tujuan; alamat_tujuan:</pre>

5.	Melompat keluar dari perulangan	<code>break;</code>
6.	Perulangan	<code>while (kondisi) { (aksi yang dikerjakan) }</code>
7.	Perulangan	<code>Do { (aksi yang dikerjakan) } while (kondisi);</code>
8.	Perulangan	<code>for (nilai_awal,syarat,operasi++/--) { (aksi yang dikerjakan) };</code>

*kondisi diisi dengan membandingkan nilai atau variabel dengan operator perbandingan.

FUNGSI DIGITAL

Fungsi digital digunakan untuk pemrograman yang menggunakan fungsi pin digital pada Arduino.

```
pinMode( pin, mode);
```

Pin adalah nomor pin yang akan digunakan. Jika menggunakan Arduino Uno, pin digitalnya berada pada pin dari 0 hingga 13. Mode menentukan pin tersebut digunakan sebagai masukan / INPUT atau keluaran / OUTPUT. Contoh:

```
pinMode(13, OUTPUT); // pin 13 digunakan sebagai OUTPUT
pinMode(7, INPUT); // pin 7 digunakan sebagai INPUT
```

Karena setting mode pin hanya cukup dilakukan satu kali saja untuk kode **pinMode** itu ada didalam **void setup**.

```
digitalRead(pin);
```

Kode di atas digunakan pin **INPUT**, untuk membaca nilai sensor yang ada pada pin dan nilainya hanya terbatas pada **1 (TRUE)**, atau **0 (FALSE)**. Contoh:

```
digitalRead(13); // artinya kode akan membaca nilai  
sensor pada pin 13
```

Kode **digitalRead** biasa di masukan dalam **void loop** karena tentu membutuhkan pembacaan kondisi secara terus menerus.

```
digitalWrite(pin, nilai);
```

Kode di atas digunakan untuk pin **OUTPUT** yang sudah kita *setting* apakah akan diberikan **HIGH (+5V)**, atau **LOW (Ground)**. Contoh:

```
digitalWrite(13, HIGH); // pin 13 diberi tegangan +5V  
digitalWrite(13, LOW); // pin 13 diberi tegangan 0 / Ground
```

Meskipun Arduino adalah perangkat digital, namun *programmer* tetap dapat membuat seolah-olah keluaran menjadi analog sehingga dapat digunakan untuk meredupkan LED maupun mengontrol kecepatan motor. Fungsi analog tersebut biasa disebut PWM. Untuk menggunakan fungsi analog pada Arduino adalah dengan perintah “analogWrite”. Format penulisan adalah sebagai berikut:

```
analogWrite(pin, nilai);
```

Arduino Uno memiliki 6 pin PWM, yaitu: 3,5,6,9, 10, dan 11. PWM pada Arduino standarnya memiliki ukuran 8bit yang membuat tegangan keluaran rata-rata (VAvg) dari tiap pin PWM menjadi bervariasi yaitu dari 0 sampai 225 atau setara dengan 0 sampai 5V.

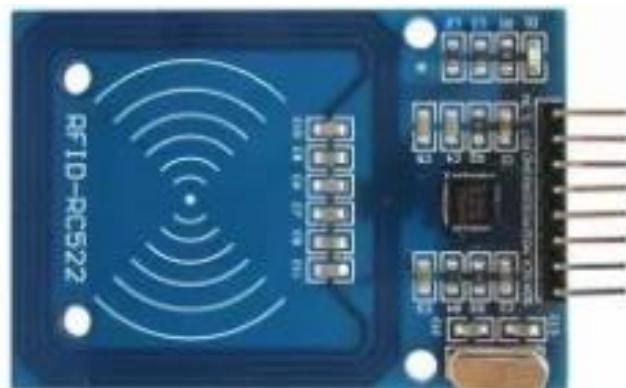
Contoh:

```
analogWrite(3, 150); // pin 3 diberikan nilai sebesar 150
```

BAGIAN II – RADIO FREQUENCY IDENTIFICATION (RFID)

A. Modul RFID RC-522

Radio Frequency Identification atau RFID pada dasarnya adalah sebuah sistem yang memungkinkan transfer data pada jarak yang dekat (Wicaksono & Hidayat, 2017:175). Konsep RFID mirip seperti *barcode*. Perbedaan yang paling mendasar adalah bahwa informasi (kode) pada *barcode* tampak secara visual, sedangkan informasi pada RFID berbentuk gelombang radio sehingga tidak tampak secara visual. Keunggulan dari RFID adalah informasi yang ada padanya bisa dibaca, ditulis, dan dikunci serta lebih awet karena lebih aman dari perlakuan fisik, sedangkan *barcode* hanya bisa dibaca serta lebih rentan dengan perlakuan fisik. Perlakuan fisik yang dimaksud adalah bahwa *barcode* dapat hilang, luntur, serta tergores sehingga informasi pada *barcode* bisa tidak terbaca.



Gambar 6. Modul RFID *reader* MFRC 522

(Sumber: <https://www.lazada.com.my/>)

RFID menggunakan gelombang radio untuk melakukan identifikasi. Tabel penggunaan frekuensi RFID dapat dilihat pada Tabel 5. Maka dari itu RFID terdiri dari dua bagian yaitu RFID *tag* sebagai penyimpan informasi atau yang dibaca, dan

pembaca yang disebut juga *RFID reader*. Saat melakukan pembacaan *RFID reader* membaca gelombang yang dikirim oleh *RFID tag*. Pada penelitian ini frekuensi *RFID* yang digunakan adalah 13,56 MHz.

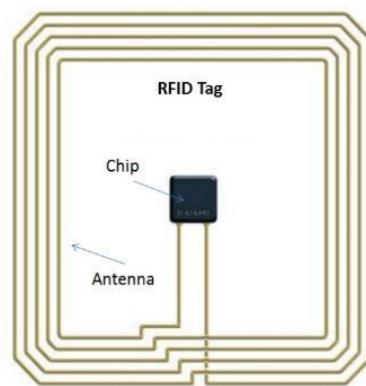
Tabel 5. Daftar Alokasi Frekuensi *RFID*

Band	Regulations	Range	Data speed	Remarks	Approximate tag cost in volume (2006) US \$
120 –150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection	\$1
13.56 MHz (HF)	ISM band worldwide	10 cm – 1 m	Low to moderate	Smart cards (MIFARE, ISO/IEC 14443)	\$0.50
433 MHz (UHF)	Short Range Devices	1–100 m	Moderate	Defence applications, with active tags	\$5
865-868 MHz (Europe) 902-928 MHz (North America) UHF	ISM band	1–12 m	Moderate to high	EAN, various standards	\$0.15 (passive tags)
2450-5800 MHz (microwave)	ISM band	1–2 m	High	802.11 WLAN, Bluetooth standards	\$25 (active tags)
3.1–10 GHz (microwave)	Ultra wide band	to 200 M	High	requires semi-active or active tags	\$5 projected

a. *Radio Frequency Identification (RFID) Tag*

RFID tag memiliki *chip* semikonduktor dengan sirkit frekuensi radio, sirkit logika, sirkit memori, dan antena yang terpasang pada media tersebut (Brady dkk., 1997:1). *RFID tag* dapat berupa perangkat aktif dan pasif. Perangkat aktif berarti *RFID tag* memiliki daya sendiri sedangkan perangkat pasif berarti *RFID* tidak

memiliki daya sendiri. *RFID tag* pasif lebih banyak digunakan karena ukurannya lebih kecil dan biaya lebih murah. *RFID tag* dapat berupa *read-only* yaitu hanya dapat dibaca maupun *read-write* yaitu bisa dibaca dan diisi kembali. *RFID tag* hampir dapat disematkan ke semua benda. Pembacaan *RFID tag* dapat dilakukan pada jarak tertentu dan tanpa terlihat. Untuk penelitian kali ini menggunakan *RFID tag* yang disematkan pada sebuah kartu identitas.



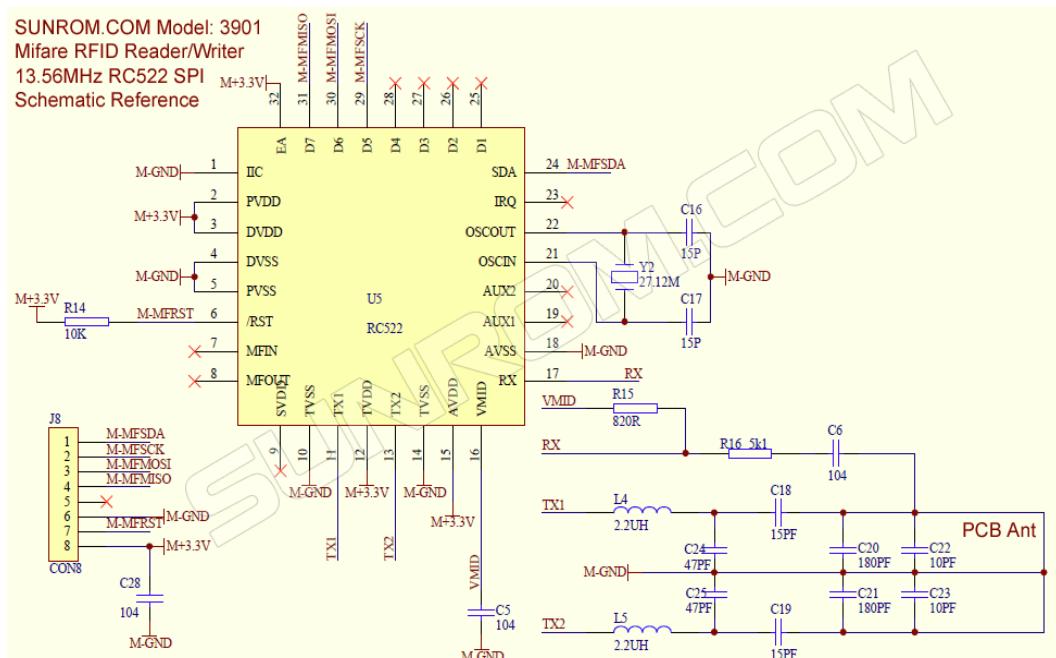
Gambar 7. Ilustrasi RFID Tag

(Sumber: <https://www.analogictips.com/>)

b. *Radio Frequency Identification (RFID) Reader*

RFID reader adalah perangkat pembaca *RFID tag*. Sama seperti *RFID tag*, *RFID reader* dapat bersifat aktif dan pasif. *RFID reader* pasif berarti tidak memiliki daya sendiri untuk melakukan pembacaan sehingga harus disuplai dari *RFID tag* aktif. Sifat tersebut membuat *RFID reader* pasif hanya dapat membaca *RFID tag* aktif. *RFID reader* aktif memiliki daya sendiri dan dapat memancarkan sinyal interogator ke *RFID tag* pasif. Sinyal interogator menginduksi *RFID tag* sehingga menjadi sumber daya pada *RFID tag* pasif tersebut. Pada penelitian ini, modul *RFID reader* yang digunakan adalah modul RC522 yang berbasis IC MFRC522.

IC MFRC522 yang berada pada modul RFID *reader* sebetulnya dapat menggunakan 3 antarmuka komunikasi yaitu *Serial Peripheral Interface* (SPI), I2C, dan UART (NXP Semiconductor, 2016). Namun pada modul RFID *reader* yang digunakan, komponen yang terpasang hanya mendukung antarmuka komunikasi SPI saja. Modul MFRC522 menggunakan komunikasi SPI karena transfer data dapat terjadi dengan cepat. Modul MFRC522 ini nantinya akan bekerja pada Arduino Nano yang bertindak sebagai *slave*. Informasi identitas yang dimiliki RFID *tag* kemudian akan dikirim menggunakan antarmuka komunikasi serial UART oleh *slave* ke *master* yaitu Arduino Mega. Secara rinci fungsi kaki-kaki pada modul RC522 dapat dilihat pada Tabel 6.



Gambar 8. Skema Modul RFID *reader* MFRC522

(Sumber: <https://www.sunrom.com/>)

Tabel 6. Fungsi kaki-kaki pada modul RFID RC522
 (Sumber: Wicaksono & Hidayat, 2017:176)

Pin	Label	Fungsi
1	SDA	Pin <i>Slave Select</i> untuk komunikasi SPI
2	SCK	Pin <i>Serial Clock</i> untuk komunikasi SPI
3	MOSI	Pin MOSI untuk komunikasi SPI
4	MISO	Pin MISO untuk komunikasi SPI
5	IRQ	tidak terhubung
6	GND	<i>Ground</i> atau 0V
7	RST	Pin Reset pada komunikasi SPI
8	3,3V	Tegangan masukan 3,3V

B. Pemrograman modul RFID dengan Arduino

Pemrograman modul RFID dengan Arduino diperlukan *library* “MFRC522.h” dan *library* komunikasi “SPI.h”. Modul RFID yang digunakan menggunakan komunikasi SPI maka dari itu, selain penyebutan *include library*, perlu pendefinisian pin SS dan RESET, deklarasi variabel, dan pengaktifan fungsi kelas MFRC522. Contoh:

```
#include <SPI.h> //memasukkan library SPI
#include <MFRC522.h> //memasukkan librari MFRC522
#define SS_PIN 10 //definisi pin SS
#define RST_PIN 9 //definisi pin RESET
MFRC522 rfid(SS_PIN, RST_PIN)//pengaktifan pin SS dan RESET
MFRC522::MIFARE_Key key;//pengaktifan fungsi kelas
byte uidPICC[4]; //deklarasi variabel
```

Supaya mempermudah pemrograman, baris pemrograman pembacaan RFID dipisah menjadi satu fungsi tersendiri. Fungsi tersebut mengandung identifikasi kartu baru, verifikasi ID yang terbaca, memecah hasil pembacaan IDE kedalam array, menampilkan/mengirimkan hasil pembacaan pada Serial Monitor, kemudian menghentikan proses pembacaan dan mengembalikan ke standby. Contoh:

```
void readRFID()
{
    // identifikasi pembacaan baru kartu rfid
    if (rfid.PICC_IsNewCardPresent())
    {
```

```

// verifikasi ID baru
if (rfid.PICC_ReadCardSerial())
{
// memecah ID hasil pembacaan kedalam array
for (byte i = 0; i < 4; i++)
{
    nuidPICC[i] = rfid.uid.uidByte[i];
}
printHex(rfid.uid.uidByte, rfid.uid.size);
// menghentikan pembacaan kemudian standby
rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}
}
}
}

```

Untuk menampilkan/mengirimkan hasil pembacaan RFID pada Serial Monitor diperlukan fungsi konversi dari pembacaan RFID ke dalam komunikasi serial.

Contoh:

```

void printHex(byte *buffer, byte bufferSize) {
    //fungsi menampilkan hasil pembacaan pada serial monitor
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? "0" : "");
        Serial.print(buffer[i], HEX);
    }
}

```

Setelah itu pada void setup diperlukan inisialisasi komunikasi SPI, Serial sebagai antarmuka komunikasi dengan pengguna/master dan pengaktifan kelas RFID dan pengecekan *hardware* modul MFRC522. Program pada **void setup** adalah sebagai berikut:

```

void setup() {
    Serial.begin(9600);
    SPI.begin(); // Init SPI bus
    rfid.PCD_Init(); // Init MFRC522
    for (byte i = 0; i < 6; i++) {
        key.keyByte[i] = 0xFF;
    }
    printHex(key.keyByte, MFRC522::MF_KEY_SIZE);
}

```

BAGIAN III – REAL TIME CLOCK (RTC)

A. Modul RTC DS3231

Real-time-clock (RTC) merupakan suatu modul yang memiliki IC (*integrated circuit*) dan baterai yang berfungsi untuk menyimpan informasi waktu dan tanggal secara terus menerus walaupun tidak mendapat pasokan daya dari luar (Suhaeb dkk., 2017:155). Salah satu IC pada modul RTC adalah DS3231. IC DS 3231 menggunakan I2C (*inter integrated circuit*) sebagai antarmuka komunikasinya. RTC memiliki keistimewaan yaitu memiliki sinyal keluaran *programable squarewave*, deteksi kegagalan daya, konsumsi daya kurang dari 500mA, serta menggunakan mode baterai untuk operasional osilator sehingga dapat tetap menyimpan informasi waktu yang akurat terus menerus. RTC memiliki ketahanan suhu dari -40 Celcius hingga +85 Celcius. Adapun fungsi tiap pin dapat dilihat pada

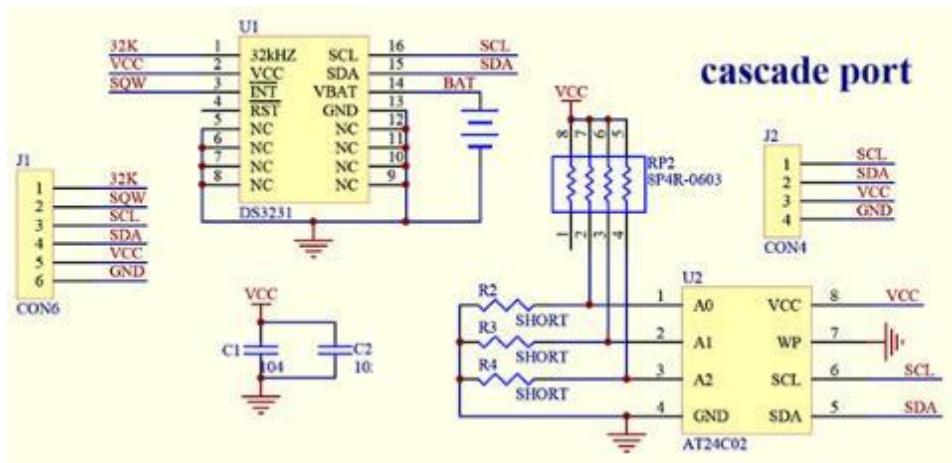
Tabel 7.

Tabel 7. Fungsi tiap pin modul RTC DS3231

Label pin	Fungsi
VCC	Sumber daya positif 5V
GND	Sumber daya negatif 0V
SDA	<i>Serial Data</i> pin (antarmuka I2C)
SCL	<i>Serial Clock</i> pin (antarmuka I2C)
SQW	<i>Square wave output</i> pin
32K	32K oscillator output.



Gambar 9. Modul *Real Time Clock* (RTC)
(Sumber: <https://edwardmallon.files.wordpress.com/>)



Gambar 10. Skema modul RTC DS3231
 (Sumber: <https://edwardmallon.files.wordpress.com/>)

B. Pemrograman RTC dengan Arduino

Jika kita membicarakan sistem pada jam, maka fungsi dasar yang harus ada pada sistem jam tersebut adalah menampilkan waktu dan sinkronisasi atau pencocokan waktu. Sama halnya seperti Modul RTC. Dasar program yang harus bisa dilakukan adalah menampilkan waktu dan melakukan pencocokan waktu. Untuk memprogram Modul RTC DS3231 pada Arduino, diperlukan *library* “DS3231.h” dan karena modul menggunakan komunikasi I2C maka diperlukan juga *library* “Wire.h”. Selain memasukkan *library*, hal yang perlu dilakukan lainnya adalah pengaktifan kelas fungsi kelas DS3231 dan pendeklarasian variabel yang diperlukan. Programnya adalah sebagai berikut:

```
#include <Wire.h>
#include <DS3231.h>
DS3231 Clock;
bool Century=false;
bool h12;
bool PM;
byte Year;
byte Month;
byte Date;
byte DoW;
```

```
byte Hour;
byte Minute;
byte Second;
```

Dasar program pertama adalah menampilkan waktu. Untuk menampilkan waktu, ada beberapa fungsi yang sudah disediakan oleh *library* “DS3231.h”.

Berikut adalah fungsi program untuk menampilkan waktu:

Tabel 8. Fungsi untuk mengambil nilai waktu dari RTC

Fungsi	Tujuan
Clock.getHour(h12, PM)	Untuk mendapat nilai jam
Clock.getMinute()	Untuk mendapat nilai menit
Clock.getSecond()	Untuk mendapat nilai detik
Clock.getDate()	Untuk mendapat nilai tanggal
Clock.getMonth(Century)	Untuk mendapat nilai bulan
Clock.getYear()	Untuk mendapat nilai tahun

Masing-masing fungsi tersebut akan menghasilkan nilai sesuai dengan tujuannya sehingga harus dimasukkan ke dalam variabel atau langsung dimasukkan/dikirimkan melalui komunikasi serial dengan perintah “Serial.print()”.

Dasar program selanjutnya adalah sinkronisasi waktu. Untuk melakukan sinkronisasi waktu, ada beberapa fungsi yang sudah disediakan oleh *library* “DS3231.h”. Berikut adalah fungsi program untuk melakukan sinkronisasi waktu.

Tabel 9. Fungsi untuk mengatur nilai waktu pada RTC

Fungsi	Tujuan	Rentang nilai
Clock.setYear(var);	Untuk menyinkronkan tahun	00 – 99
Clock.setMonth(var);	Untuk menyinkronkan bulan	01 – 12
Clock.setDate(var);	Untuk menyinkronkan tanggal	01 – 30
Clock.setDoW(var);	Untuk menyinkronkan hari	01 – 07
Clock.setHour(var);	Untuk menyinkronkan jam	00 – 23
Clock.setMinute(var);	Untuk menyinkronkan menit	00 – 59
Clock.setSecond(var);	Untuk menyinkronkan detik	00 – 59

Hal yang perlu diperhatikan selanjutnya adalah membuat sintak program yang dapat digunakan sebagai antarmuka pengguna untuk melakukan sinkronisasi

waktu. Antarmuka pengguna tersebut bisa menggunakan *keypad* atau perintah dari Serial Monitor. Berikut adalah contoh sintak fungsi jika menggunakan perintah dari Serial Monitor. Fungsi diberi nama “GetDateStuff”.

```
void GetDateStuff(byte& Year, byte& Month, byte& Day, byte&
DoW, byte& Hour, byte& Minute, byte& Second)
{
    /* fungsi ini digunakan untuk melakukan parsing data
    dari perintah. Untuk melakukan sinkronisasi waktu adalah
    dengan mengirimkan string dari serial monitor dengan format:
    YYMMDDwHHMMSS, dengan 'x' pada akhirannya.
    misal: 190213w130900x */
    boolean GotString = false;
    char InChar;
    byte Temp1, Temp2;
    char InString[20];
    byte j=0;
    while (!GotString) {
        if (Serial.available()) {
            InChar = Serial.read();
            InString[j] = InChar;
            j += 1;
            if (InChar == 'x') { GotString = true; }
        }
    }
    Serial.println(InString);
    //berikut adalah parsing data dari pembacaan serial
    monitor yang sudah dikirim.
    // parsing baca tahun
    Temp1 = (byte)InString[0] -48;
    Temp2 = (byte)InString[1] -48;
    Year = Temp1*10 + Temp2;
    // parsing baca bulan
    Temp1 = (byte)InString[2] -48;
    Temp2 = (byte)InString[3] -48;
    Month = Temp1*10 + Temp2;
    // parsing baca tanggal
    Temp1 = (byte)InString[4] -48;
    Temp2 = (byte)InString[5] -48;
    Day = Temp1*10 + Temp2;
    // parsing Day of Week
    DoW = (byte)InString[6] - 48;
    // parsing baca jam
```

```

Temp1 = (byte)InString[7] -48;
Temp2 = (byte)InString[8] -48;
Hour = Temp1*10 + Temp2;
// parsing baca menit
Temp1 = (byte)InString[9] -48;
Temp2 = (byte)InString[10] -48;
Minute = Temp1*10 + Temp2;
// parsing baca detik
Temp1 = (byte)InString[11] -48;
Temp2 = (byte)InString[12] -48;
Second = Temp1*10 + Temp2;
}

```

Fungsi untuk mengambil dan menyinkronkan waktu tentu disesuaikan dengan kebutuhan. Misalkan saja jika ingin menampilkan waktu pada LCD akan berbeda dengan menampilkan waktu pada 7-Segment. Begitu juga untuk menyinkronkan waktu. Fungsi menyinkronkan dari serial monitor tentu berbeda jika akan memasukkan dari *keypad*. Namun yang perlu diingat bahwa *library* RTC sudah menyediakan fungsi-fungsi untuk pemanggilan nilai maupun pengaturan nilai. Semuanya sudah berbentuk variabel sehingga lebih mudah dan fleksibel diaplikasikan dalam berbagai perangkat. Berikut adalah contoh program jika nilai jam akan ditampilkan dalam LCD:

```

void showClock(bool a)
{
    lcd.setCursor(0,a);
    lcd.print(Clock.getHour(h12, PM));
    if((Clock.getSecond() % 2) == 0) lcd.print(":");
    else lcd.print(" ");
    lcd.print(Clock.getMinute());
    lcd.print(" ");
    lcd.print(Clock.getDate());
    lcd.print("/");
    lcd.print(Clock.getMonth(Century));
    lcd.print("/");
    lcd.print(Clock.getYear());
    lcd.print("  ");
}

```

BAGIAN IV – MODUL MICRO SD CARD

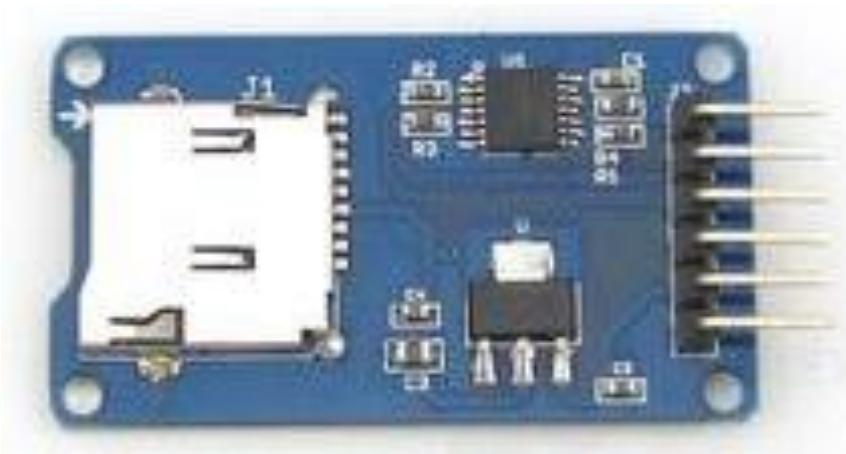
A. Modul MicroSD Card Reader-Writer

Modul (MicroSD Card Adapter) adalah modul untuk membaca dan menulis data pada kartu memori mikro yang menggunakan antarmuka SPI (Faudin, 2018). Modul ini cocok digunakan untuk aplikasi yang membutuhkan penyimpanan basis data seperti sistem presensi, antrean, *data logging*, sistem parkir, dan sebagainya. Secara rinci fungsi kaki-kaki pada modul RFID dapat dilihat pada Tabel 10.

Tabel 10. Fungsi kaki-kaki pada modul *micro sd card* adapter

(Sumber: Agus Faudin, 2018)

Pin	Label	Fungsi
1	CS	Pin <i>Slave Select</i> untuk komunikasi SPI
2	SCK	Pin <i>Serial Clock</i> untuk komunikasi SPI
3	MOSI	Pin MOSI untuk komunikasi SPI
4	MISO	Pin MISO untuk komunikasi SPI
5	VCC	Sumber daya positif 5V
6	GND	Sumber daya negatif 0V



Gambar 11. Modul SD Card Adapter
(Sumber: <http://www.indo-ware.com>)

B. Pemrograman Modul Micro SD dengan Arduino

Pemrograman Modul Micro SD dengan Arduino diperlukan *library* “SD.h” dan karena modul tersebut menggunakan komunikasi SPI maka *library* yang digunakan selanjutnya adalah *library* komunikasi SPI yaitu “SPI.h”. Seperti halnya komputer, dengan Modul Micro SD Arduino dapat melakukan manajemen file seperti membuka, membuat, dan menghapus file maupun folder.

Library “SD.h” memiliki 2 kelas yaitu kelas **SD Class** dan **File Class**. SD Class memiliki fungsi untuk mengakses MicroSD dan memanipulasi file dan folder. *File Class* berfungsi untuk melakukan baca tulis dalam sebuah file. Ekstensi file yang dapat diakses dengan *library* ini adalah teks atau “txt” ataupun tanpa ekstensi namun berformat teks.

Tabel 11. Fungsi untuk mengakses micro SD

Nama	Fungsi	Sintak
begin()	Memulai pembacaan	SD.begin(PINSS)
exists()	Cek keberadaan file atau folder	SD.exists(namafolder)
mkdir()	Membuat folder	SD.mkdir(namafolder)
open()	Membuka file	SD.open(filepath, mode) mode=“FILE_READ” untuk membuka file. mode=“FILE_WRITE” untuk membuat file.
remove()	Menghapus file atau folder	SD.remove(filename)
rmdir()	Menghapus folder	SD.rmdir(foldername) *folder harus kosong

Untuk memulai memprogram MicroSD, hal yang perlu dilakukan adalah menyebutkan library MicroSD dan library antarmuka komunikasinya. Library MicroSD adalah “SD.h” dan library SPI adalah “SPI.h” Sedangkan untuk memulai

pembacaan file, harus mengaktifkan kelas terlebih dahulu. Contoh *header* pemrograman MicroSD adalah sebagai berikut:

```
#include <SPI.h> //library komunikasi SPI
#include <SD.h> //library modul SDCARD
File myFile; //inisialisasi file pada SDCard
```

Bagian void setup berisi pengecekan apakah MicroSD sudah tersambung atau belum. Programnya adalah sebagai berikut:

```
if (!SD.begin(53)) // Pengecekan MicroSD dengan SS pada pin 53
{
  while (1){}; //program akan terkunci jika tidak sukses
  else
  {
    /* tindakan program jika sukses */
  }
}
```

Jika sudah sukses, maka Anda bisa mulai melakukan manajemen data pada MicroSD. Misalnya adalah pencarian dalam *database* atau tulis menulis data. Berikut adalah contoh program pencarian dalam *database*:

```
String scanDatabase(String a="")
{
  //ini adalah fungsi untuk mencari ID yang sudah terdaftar
  String result="";
  String database="";
  unsigned int found=0, nameStart=0;
  myFile = SD.open("DATABASE/DATABASE.txt");
  if (myFile)
  {
    while (myFile.available())
    {
      database=myFile.readString();
      found=database.indexOf(a);
      if(found!=-1)
      {
        nameStart=database.lastIndexOf("\n", found);
        result=database.substring(nameStart+1,found-1);
      }
    }
  }
}
```

```

    {
        result="NOTFOUND";
    }
    found=0;
}
myFile.close();
}
else
{
    result="DATABASEERROR";
}
return result;
}

```

Program di atas adalah contoh program pencarian ID pada database. Program tersebut dapat berfungsi untuk berbagai keperluan. Misalkan saja pencarian ID pada sistem parkir, pencarian member tertentu, dan sebagainya. Selain melakukan pencarian *database*, fungsi yang tidak kalah penting adalah melakukan pembacaan dan pembuatan file. Berikut adalah contoh program penulisan file.

```

void writeSetting(unsigned int userX,unsigned int quotaX,unsigned int
timeN,unsigned int timeX)
{
    //ini adalah fungsi untuk mengubah pengaturan
    if (SD.exists("DATABASE/SETTING.txt"))
    {
        SD.remove("DATABASE/SETTING.txt");
        myFile = SD.open(String("DATABASE/SETTING.txt"), FILE_WRITE);
        myFile.print("KALIMAT YANG INGIN Anda TULISKAN DALAM SETTING.TXT")
        myFile.close()
    }
    else
    {
        while(1){} //JIKA FILE SETTING.txt tidak tersedia
    }
}

```

BAGIAN V – MOTOR DC

A. *Brushed* Motor DC

Motor DC adalah komponen yang mengubah energi listrik menjadi energi gerak. Prinsip kerja motor DC menggunakan prinsip induksi magnetik. Sirkuit internal motor DC terdiri dari lilitan konduktor. Arus mengalir di setiap konduktor dan menimbulkan medan magnet. Konduktor didesain menjadi sebuah *loop* sehingga ada dua bagian konduktor yang berada pada medan magnet yang sama di saat yang sama (Wicaksono & Hidayat, 2017: 141). Motor DC dapat berputar 360 derajat secara terus menerus sesuai arah jarum jam maupun berlawanan arah jarum jam. Putaran Motor DC dapat dibalik dengan mengubah polaritas kutub-kutubnya (Andrianto & Darmawan, 2017:131).



Gambar 12. Motor DC dengan *gearbox*

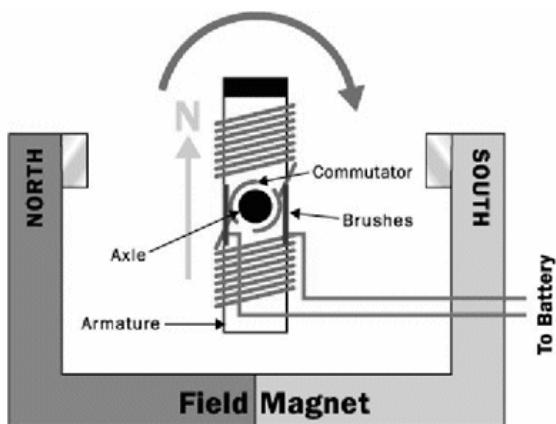
(Sumber: <https://www.tokopedia.com/lek-electronics/>)

Pada penelitian ini motor dc digunakan untuk mekanisme membuka dan menutup palang. Penelitian ini menggunakan motor DC dengan *gearbox* karena memiliki torsi yang lebih besar dari pada motor DC tanpa *gearbox* dengan spesifikasi motor DC yang sama. Motor DC dengan *gearbox* memiliki energi

kinetik yang besar. Energi kinetik yang besar menyebabkan poros motor dc membutuhkan waktu yang cukup lama hingga berhenti. Hal tersebut akan menjadi kerugian jika sistem membutuhkan gerakan berhenti motor yang presisi. Untuk mengatasi hal tersebut diperlukan sistem penggereman. Sistem penggereman motor dc dapat dilakukan dengan menghubungkan kutub-kutub motor dc.

Elemen utama motor DC adalah:

- Magnet
- Armatur atau rotor
- Komutator
- Sikat (*Brushes*)
- As atau poros (*Axle*)



Gambar 13. Visualisasi cara kerja motor DC

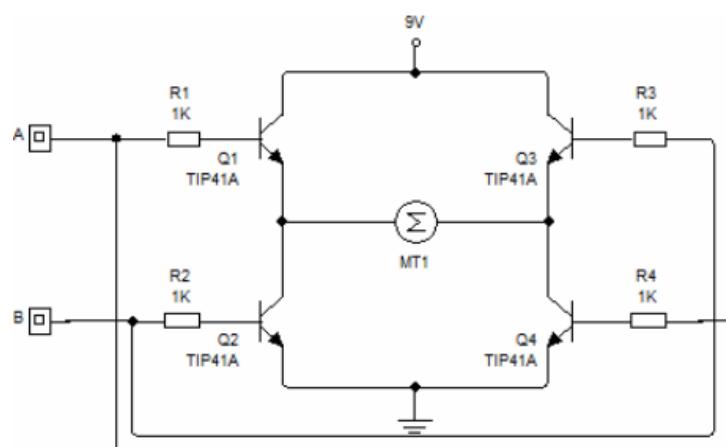
(Sumber: <http://www.mikron123.com>)

B. Mengendalikan *Brushed* Motor DC

Untuk menghidupkan motor DC cukup dengan memberi dia beda potensial di tiap kutubnya. Jika beda potensial dibalik maka putaran motor dc juga akan kebalikannya. Aktivitas pengendaian seperti memutar dan membalik putaran bisa

menggunakan saklar mekanik, namun jika ingin mengendalikan motor dc menggunakan mikrokontroller atau Arduino diperlukan sebuah rangkaian driver tertentu. Banyak rangkaian driver yang beredar di Internet maupun buku-buku tentang elektro dan elektronika. Rangkaian kendali motor dc yang paling populer adalah rangkaian h-bridge.

Rangkaian h-bridge adalah rangkaian yang terdiri dari 4 saklar mekanik maupun elektronik. Rangkaian ini dapat disusun menggunakan 4 transistor. Rangkaian ini dapat melakukan fungsi yaitu memutar motor, membalik putaran motor, dan mengatur kecepatan motor. Untuk mengendalikan dengan rangkaian h-bridge transistor, diperlukan dua masukan yaitu A dan B. Untuk menjalankan motor, sinyal A dan B harus berbeda misal A HIGH atau satu (1) berarti B LOW atau nol (0) dan jika ingin membalik putarannya maka tinggal dibalik saja. Rangkaian h-bridge tidak bisa bekerja jika masukan A dan B semuanya LOW atau semuanya HIGH. Untuk mengatur kecepatan motor maka sinyal A dan B bisa dikendalikan dengan sinyal *pulse-width modulation* (PWM).



Gambar 14. Rangkaian H-Bridge dengan 4 transistor

Selain rangkaian h-bridge, ada banyak rangkaian kendali motor dc. Bahkan sudah banyak tersedia IC seperti L293D, DRV8802, MAX14872 dan masih banyak lagi. Bahkan jika hanya membutuhkan pengendalian menyala-mati dan tidak memerlukan perubahan kecepatan, maka cukup menggunakan satu transistor maupun satu relai saja. Perancang sistem tentu harus pandai memilih cara pengendalian yang sesuai dengan kebutuhan agar alat yang dibuat menjadi lebih efektif dan efisien dalam hal program maupun biaya.

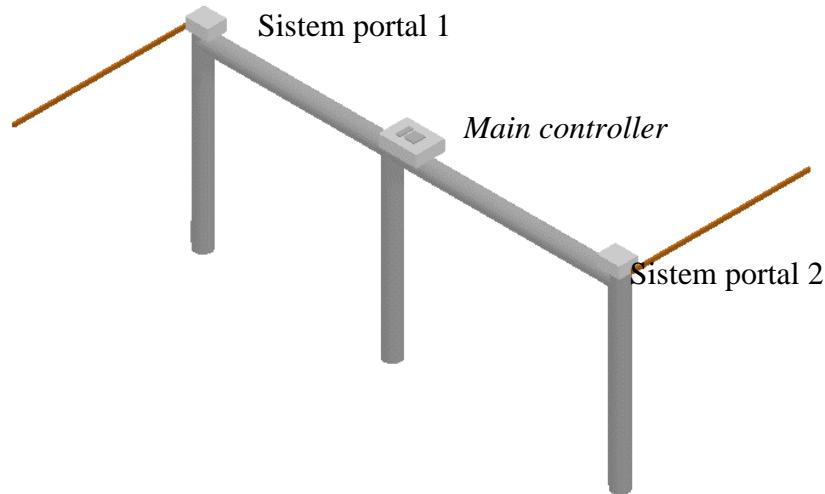
Bahkan jika dibutuhkan perancang sistem dapat mendesain sendiri motor driver. Misalkan saja motor dc untuk keperluan buka tutup portal. Untuk keperluan buka-tutup portal, motor dc yang diperlukan adalah memiliki torsi besar dan membalik-balikkan putaran.

BAGIAN VI – SISTEM PENGATURAN PARKIR BERBASIS RFID

A. Pengenalan Sistem

Sistem Pengaturan Parkir berbasis RFID ini adalah suatu sistem pengelolaan parkir yang digunakan pada fasilitas parkir khusus. Sistem Pengaturan Parkir ini menggunakan kartu identitas khusus yang diberikan kepada setiap pengguna parkir. Kartu identitas ini menggunakan teknologi RFID. Sistem ini memiliki sistem pengamanan dan pembatasan gerak pengguna. Sistem pengamanan berupa pemberian hak parkir hanya pengguna yang memiliki kartu parkir dan terdaftar dalam sistem. Pembatasan gerak tersebut berupa waktu parkir dan kuota parkir.

Sistem parkir ini menggunakan mikrokontroller sebagai pemrosesan utamanya. Sistem parkir ini memiliki tiga bagian. Bagian pertama adalah *main controller*, bagian kedua dan ketiga adalah sistem portalnya. *Main controller* berisi kontroler utama berupa sistem *master-slave* Arduino Mega dan Arduino Nano, pembaca RFID, layar informasi, serta *keypad*. Sistem portal berisi *motor dc*, *driver motor dc*, serta sensor ultrasonik. Motor dc beserta *driver*-nya digunakan sebagai sistem penggerak palang sedangkan sensor ultrasonik digunakan sebagai sensor lewat kendaraan. Antara *main controller* dengan sistem portal dihubungkan melalui koneksi berkabel (*wired connection*).



Gambar 15. Desain Posisi *Main Controller* dengan Sistem Portal

Boks *main controller* berukuran panjang 20cm, lebar 15cm, dan tinggi 8 cm.

Boks sistem portal memiliki ukuran panjang 12cm, lebar 10cm, tinggi 9cm, dan panjang portal 100cm. Masing-masing boks berbahan akrilik yang memiliki tebal 5mm dan palang berbahan dasar aluminium. Spesifikasi dari sistem pengaturan parkir berbasis RFID adalah sebagai berikut:

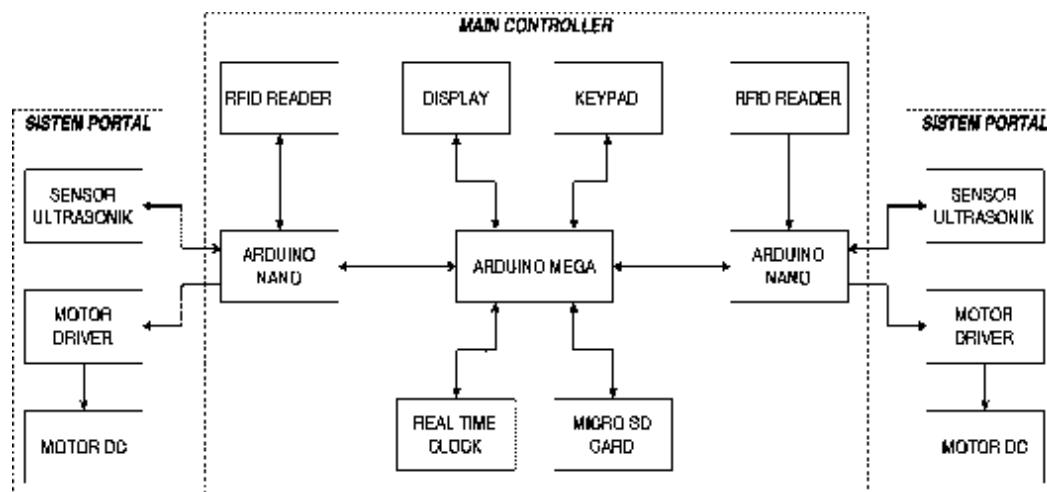
1. *Hardware*:

- a. Arduino Mega sebagai *master*
- b. Arduino Nano sebagai *slave*,
- c. Pembaca RFID dan kartu RFID sebagai simulasi pengguna parkir,
- d. *Real-time-clock* sebagai simulasi pembatasan waktu parkir,
- e. *Limit switch*, sensor *ultrasonic* dan motor dc sebagai miniatur palang pada pintu masuk parkir.
- f. Modul microSD sebagai penyimpan basis data,
- g. *Liquid Crystal Display (LCD)* sebagai penyampai informasi alfa-numerik,
- h. *Keypad* sebagai opsi masukan,

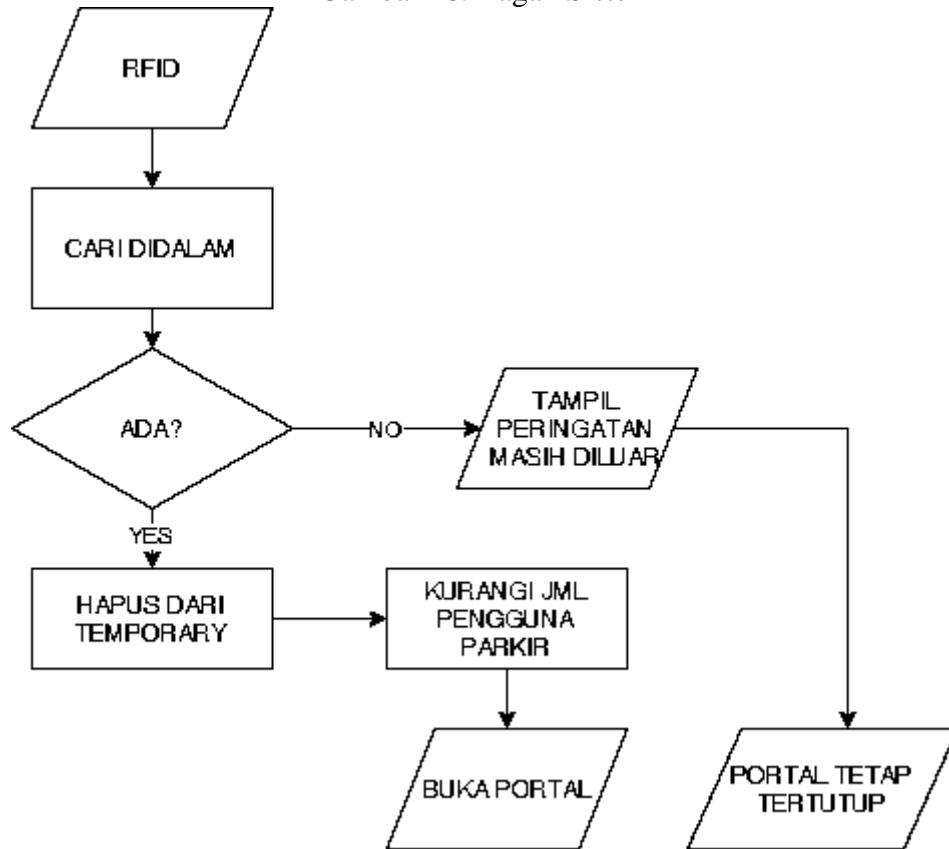
2. Software:

- a. Arduino IDE sebagai tempat penulisan serta *compiler* program pada Arduino.

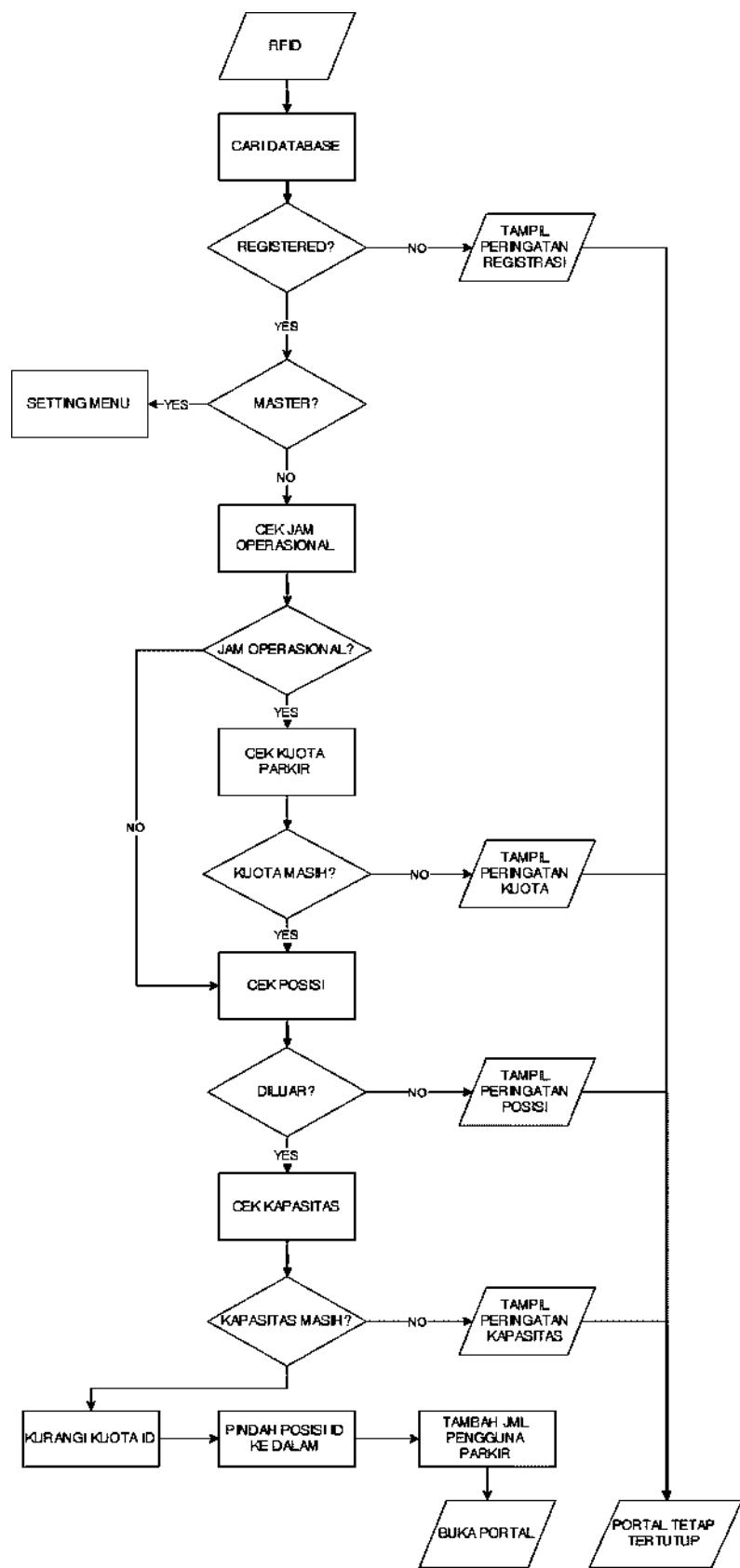
B. Bagan dan Flowchart Sistem



Gambar 16. Bagan Sistem



Gambar 17. Flowchart saat pengguna keluar



Gambar 18. Flowchart saat pengguna masuk

C. Bagian-bagian unit



Gambar 19. Bentuk fisik *main controller* dengan keterangan



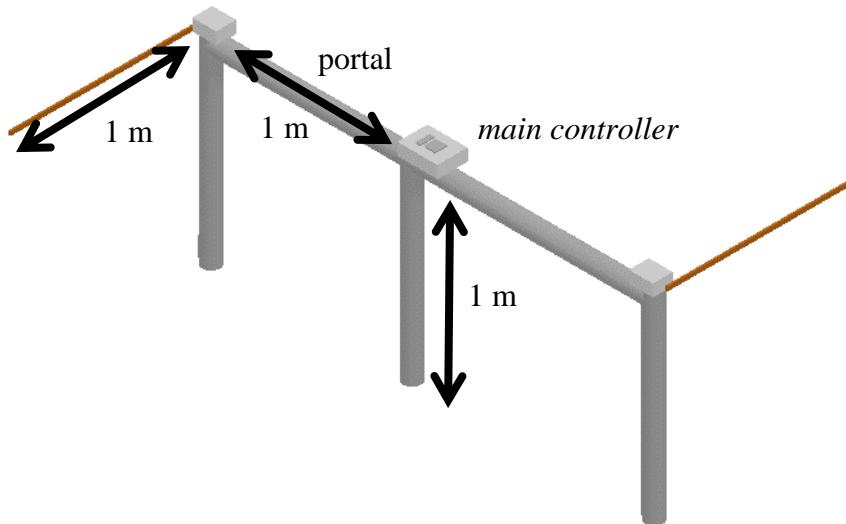
Gambar 20. Bentuk fisik bagian portal dengan keterangan



Gambar 21. Kabel konektor dan Adaptor

D. Perakitan

Sebelum proses perakitan, ada hal yang perlu diperhatikan yaitu desain tempat keluar dan masuk kendaraan. Pastikan jalan memiliki lebar 1 meter, tinggi 1 meter dan jarak antara *main controller* dan portal juga 1 meter. Selanjutnya adalah dudukan dari *main controller* serta portal, pastikan dudukan tersebut baik dan kokoh. Jalan keluar masuk kendaraan sebaiknya juga diberi atap agar sistem tidak terganggu oleh cuaca. Letakkan portal dan *main controller* dengan baik kemudian hubungkan dengan kabel konektor yang sudah tersedia. Hubungkan konektor adaptor ke unit namun jangan diberi daya terlebih dahulu sebelum melihat sub bagian “Pengoperasian”.



Gambar 22. Rekomendasi posisi unit

E. Pengoperasian

Untuk memulai pengoperasian sistem ini, Anda harus membuat file *database*, dan *setting* terlebih dahulu pada SDCard. Buat folder dengan nama “DATABASE” didalam folder tersebut, buatlah file “DATABASE.TXT” dan “SETTING.TXT”.

Pada file “DATABASE.TXT” isikan teks berikut:

```
/*
*INI ADALAH DATABASE SISTEM PARKIR
*FORMAT DATABASE = NAMAPEMILIK-RFID
*CONTOH = ARI_SANTOSO XX494FXX
*/
MASTER 8C59EF49
MASTER 6BAC95B9
ARI 49A63B2D
BUDI 24D2E920
CECEP 49FB852D
DENI 395A612D
ENI 497FE52D
FANI 1408B120
GHANI 399B362D
HANI 493A1F2D
ISNAN 39F6172D
JANI 49DBA72D
KIKI 39245E2D
LALA 39D6BC2D
MERI 49B80F2D
NENI 4977A82D
```

File “SETTING.TXT” isikan teks dengan format “(kapasitas parkir)A(kuota parkir)B(awal jam operasional)C(akhir jam operasional)D”, misal jika kapasitas parkir adalah 5, kuota parkir adalah 2, dan jam operasional dari 06-18 maka isi teks adalah “5A2B6C18D”. Isi file “SETTING.txt” dapat diganti melalui *master setting*. Setelah itu masukkan SDCard pada sistem dan hubungkan sumber daya. Untuk menggunakan sistem, pengguna cukup menempelkan kartu yang sudah ter-registrasi pada *database* dan sistem akan bekerja dengan semestinya.

Sistem ini dilengkapi dengan fungsi *master setting*. Fungsi ini dapat dilakukan pada kartu yang bernama “MASTER” pada *database*. Untuk masuk ke fungsi ini, cukup menempelkan kartu master pada RFID masuk. Setelah itu akan muncul tampilan “MASTER SETTING” pada LCD. Master setting memiliki beberapa menu yang dapat dilihat pada tabel. Untuk memilih menu cukup menekan tombol angka sesuai menu yang tersedia.

Tabel 12. Menu pada *master setting*

No	Nama Menu	Fungsi
1	BUKA PORTAL	Membuka portal masuk maupun keluar
2	SET WAKTU	Mencocokkan waktu sistem
3	SET KUOTA MAKS	Mengatur jumlah kuota maksimal pengguna
4	SET KAPASITAS	Mengatur jumlah kapasitas parkir
5	SET JAM KERJA	Mengatur jam kerja

DAFTAR PUSTAKA

- Andrianto, H., & Darmawan, A. (2017). *Arduino: Belajar Cepat dan Pemrograman*. Bandung: Informatika.
- Brady, M. J., Brewster, Cofino, T., Heinrich, H. K., Johnson, G. W., Maskowirz, P. A., & Walker, G. F. (1997). *Radio Frequency Identification Tag*. New York.
- Faudin, A. (2018). Cara Mengakses Module Micro SD Menggunakan Arduino. Diambil 15 Januari 2019, dari <https://www.nyebarilmu.com/cara-mengakses-module-micro-sd-menggunakan-arduino/>
- NXP Semiconductor. (2016). *MFRC522*. Eindhoven, Netherlands.
- Suhaeb, S., Djawad, Y. A., Jaya, H., Ridwansyah, Sabran, & Risal, A. (2017). *Mikrokontroler dan Interface*. Makassar: UNM.
- Wicaksono, M. F., & Hidayat. (2017). *Mudah Belajar Mikrokontroler Arduino*. Bandung: Informatika.

LAMPIRAN

PETUNJUK PENGGUNAAN
SISTEM PENGATURAN PARKIR BERBASIS
RADIO FREQUENCY IDENTIFICATION (RFID)

Daftar Isi

COVER	I
DAFTAR ISI	II
DAFTAR TABEL	III
DAFTAR GAMBAR	IV
A. BAGIAN-BAGIAN UNIT	1
B. PERAKITAN	2
C. PENGOPERASIAN	6

Daftar Tabel

Tabel 1. Menu pada <i>master setting</i>	7
--	---

Daftar Gambar

Gambar 1. Bentuk fisik <i>main controller</i> dengan keterangan	1
Gambar 2. Bentuk fisik bagian portal dengan keterangan	1
Gambar 3. Kabel konektor dan Adaptor	2
Gambar 4. Rekomendasi posisi unit.....	3

A. BAGIAN-BAGIAN UNIT



Gambar 1. Bentuk fisik *main controller* dengan keterangan



Gambar 2. Bentuk fisik bagian portal dengan keterangan

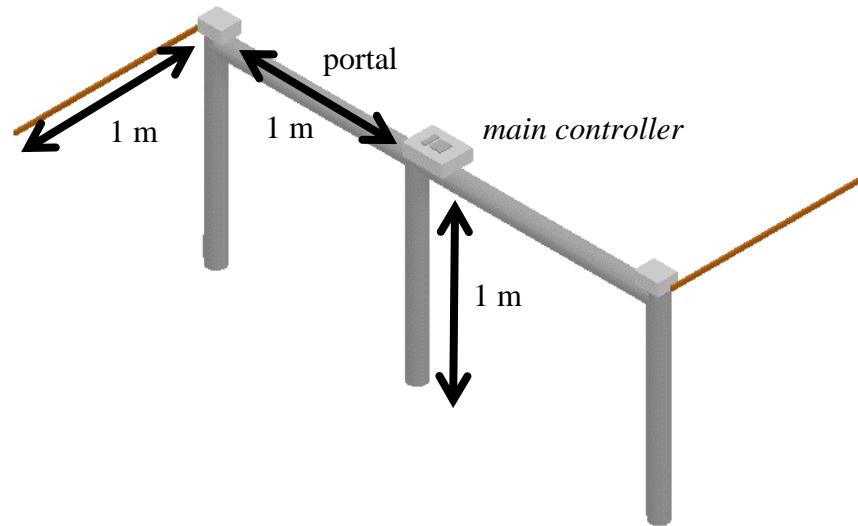


Gambar 3. Kabel konektor dan Adaptor

B. PERAKITAN

1. Pasang unit di tempat yang sesuai.

Sebelum proses perakitan, ada hal yang perlu diperhatikan yaitu desain tempat keluar dan masuk kendaraan. Pastikan jalan memiliki lebar 1 meter, tinggi 1 meter dan jarak antara *main controller* dan portal juga 1 meter. Selanjutnya adalah dudukan dari *main controller* serta portal, pastikan dudukan tersebut baik dan kokoh. Jalan keluar masuk kendaraan sebaiknya juga diberi atap agar sistem tidak terganggu oleh cuaca. Letakkan portal dan *main controller* dengan baik kemudian hubungkan dengan kabel konektor yang sudah tersedia. Hubungkan konektor adaptor ke unit namun jangan diberi daya terlebih dahulu sebelum melihat sub bagian “Pengoperasian”.



Gambar 4. Rekomendasi posisi unit

2. Rakit antar komponen



2. Pasang portal, jepitkan, kemudian kencangkan baut



3. Geser bagian ini untuk menyesuaikan ketinggian portal saat tertutup dan terbuka

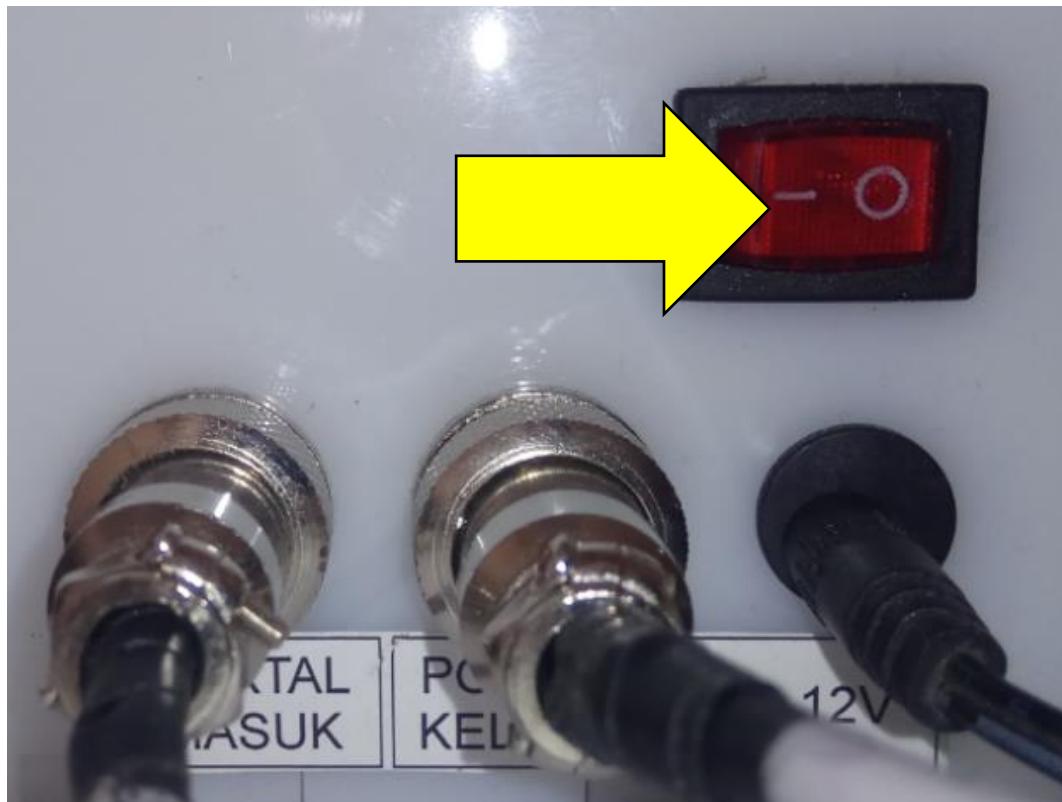




Lakukan tahap 1-4 pada gambar untuk kedua portal.



Jika dirasa sudah siap, silakan hubungkan adaptor ke stop kontak dan tekan saklar untuk menghidupkan sistem.



C. PENGOPERASIAN

Untuk mengoperasikan sistem ini, pengguna hanya perlu menempelkan kartu yang sudah teregistrasi pada sistem. Tap masuk ada di sebelah kiri, dan tap keluar ada di sebelah kanan. Sistem ini dilengkapi dengan fungsi *master setting*. Fungsi ini dapat dilakukan pada kartu yang bernama “MASTER” pada *database*. Untuk masuk ke fungsi ini, cukup menempelkan kartu master pada RFID masuk. Setelah itu akan muncul tampilan “MASTER SETTING” pada LCD. Master setting memiliki beberapa menu yang dapat dilihat pada tabel. Untuk memilih menu cukup menekan tombol angka sesuai menu yang tersedia.

Tabel 1. Menu pada *master setting*

No	Nama Menu	Fungsi
1	BUKA PORTAL	Membuka portal masuk maupun keluar
2	SET WAKTU	Mencocokkan waktu sistem
3	SET KUOTA MAKS	Mengatur jumlah kuota maksimal pengguna
4	SET KAPASITAS	Mengatur jumlah kapasitas parkir
5	SET JAM KERJA	Mengatur jam kerja

Master setting memungkinkan master/petugas parkir untuk melakukan fungsi spesial. Fungsi spesial dapat dilihat pada Tabel 1. Berikut adalah langkah-langkah untuk tiap fungsi spesial tersebut.

1. BUKA PORTAL

Tap *master card* sehingga muncul tampilan berikut



Tekan angka 1 pada *keypad* hingga muncul tampilan berikut



Kemudian tekan *enter* hingga muncul tampilan berikut



Seharusnya portal sudah terbuka pada tahap ini, jika tidak maka hubungi teknisi.

Tekan *exit* (x) untuk menutup portal

2. SET WAKTU

Tap *master card* sehingga muncul tampilan berikut



Tekan angka 2 pada *keypad* hingga muncul tampilan berikut



Kemudian tekan *enter* hingga muncul tampilan berikut



Masukkan waktu yang diinginkan dengan format YYMMDDHHMM. YY adalah 2 digit terakhir tahun, MM adalah dua digit bulan, DD adalah dua digit tanggal, HH adalah dua digit jam dalam 24 jam, dan MM adalah dua digit menit. Misal Anda ingin memasukkan waktu 1 Januari 2019 jam 08.00 maka tulis "1901010800" jika sudah maka tekan *enter*.



Jika berhasil maka akan tampil waktu seperti di bawah ini:



Jika muncul pesan *error*, hubungi teknisi.

3. SET KUOTA MAKs

Tap *master card* sehingga muncul tampilan berikut



Tekan angka 3 pada *keypad* hingga muncul tampilan berikut



Kemudian tekan *enter* hingga muncul tampilan berikut



Isikan jumlah kuota keluar masuk tiap pengguna. Misal 5 kali.



Kemudian tekan *enter*. Jika berhasil maka akan tampil seperti berikut.



Jika muncul pesan *error*, hubungi teknisi.

4. SET KAPASITAS

Tap *master card* sehingga muncul tampilan berikut



Tekan angka 4 pada *keypad* hingga muncul tampilan berikut



Kemudian tekan *enter* hingga muncul tampilan berikut



Isikan kapasitas maksimal parkir. Misal 100.



Kemudian tekan *enter*. Jika berhasil maka akan tampil seperti berikut.



Jika muncul pesan *error*, hubungi teknisi.

5. SET JAM KERJA

Tap *master card* sehingga muncul tampilan berikut



Tekan angka 5 pada *keypad* hingga muncul tampilan berikut



Kemudian tekan *enter* hingga muncul tampilan berikut



Masukkan waktu jam kerja penghitungan kuota parkir yang diinginkan (di luar jam ini, kuota tidak akan dihitung) dengan format HHHH. HH digit awal adalah jam mulai, dan HH digit akhir adalah jam selesai. Misal Anda ingin memasukkan waktu dari jam 06.00 hingga 19.00 maka tulis "0619" jika sudah maka tekan *enter*.



Kemudian tekan *enter*. Jika berhasil maka akan tampil seperti berikut.



Jika muncul pesan *error*, hubungi teknisi.